

Declaration

**Automatic classification of Spoken
South African English Variants using
a transcription-less speech
recognition approach**

André du Toit



Thesis presented in partial fulfilment of the requirements for the degree of Master of
Electronic Engineering at the University of Stellenbosch.

Prof. J.A. du Preez

2004 April

Declaration

I, the undersigned, hereby declare that the work contained in this thesis is my own original work and that I have not previously in its entirety or in part submitted it at any university for a degree.

Signature:

Date:

Abstract

We present the development of a pattern recognition system which is capable of classifying different Spoken Variants (SVs) of South African English (SAE) using a transcription-less speech recognition approach. Spoken Variants (SVs) allow us to unify the linguistic concepts of accent and dialect from a pattern recognition viewpoint. The need for the SAE SV classification system arose from the multi-linguality requirement for South African speech recognition applications and the costs involved in developing such applications.

Opsomming

Ons beskryf die ontwikkeling van 'n patroon herkenning stelsel wat in staat is om verskillende Gesproke Variante (GVe) van Suid Afrikaanse Engels (SAE) te klassifiseer met behulp van 'n transkripsielose spraak herkenning metode. Gesproke Variante (GVe) stel ons in staat om die taalkundige begrippe van aksent en dialek te verenig vanuit 'n patroon herkenning oogpunt. Die behoefte aan 'n SAE GV klassifikasie stelsel het ontstaan uit die meertaligheid vereiste vir Suid Afrikaanse spraak herkenning stelsels en die koste verbonde aan die ontwikkeling van sodanige stelsels.

Acknowledgements

I would like to thank all the people who have had a positive influence on my life; whether it was through guidance or friendship. I would rather not attempt to list all of you, in case I forget to mention anyone in particular.

1	Introduction	1
1.1	Motivation	1
1.1.1	Multi-linguality	1
1.1.2	Spoken Variants (SVs)	2
1.1.3	South African English (SAE)	2
1.1.4	Speech transcriptions	2
1.2	Objectives	3
1.3	Contributions	3
1.4	Overview	4
2	Literature synopsis	14
2.1	Feature representations used for LC recognition	14
2.1.1	Spectral speech features	14
2.1.2	Units of speech (phoneme, word etc.)	15
2.1.3	Prosodic features	15
2.2	Speech recognition techniques employed for LC recognition	15
2.2.1	LC recognition utilising spectral speech features	15
2.2.2	LC recognition utilising units of speech (phoneme, word etc.)	18
2.3	Conclusions drawn from the literature synopsis	19
3	The AST South African English (SAE) corpus	25
3.1	The SAE SVs	26
3.2	SAE corpus speech collection	27
3.3	SAE corpus speech statistics	30
4	Theory	32
4.1	Speech preprocessing	32

Contents

List of acronyms	xvi
List of symbols	xix
1 Introduction	1
1.1 Motivation	1
1.1.1 Multi-linguality	1
1.1.2 Spoken Variants (SVs)	2
1.1.3 South African English (SAE)	3
1.1.4 Speech transcriptions	4
1.2 Objectives	5
1.3 Contributions	5
1.4 Overview	6
2 Literature synopsis	14
2.1 Feature representations used for LC recognition	14
2.1.1 Spectral speech features	14
2.1.2 Units of speech (phoneme, word etc.)	15
2.1.3 Prosodic features	15
2.2 Speech recognition techniques employed for LC recognition	15
2.2.1 LC recognition utilising spectral speech features	15
2.2.2 LC recognition utilising units of speech (phoneme, word etc.)	18
2.3 Conclusions drawn from the literature synopsis	24
3 The AST South African English (SAE) corpus	26
3.1 The SAE SVs	26
3.2 SAE corpus speech collection	27
3.3 SAE corpus speech statistics	30
4 Theory	32
4.1 Speech preprocessing	32

4.1.1	Preemphasis	33
4.1.2	Power normalisation	33
4.1.3	Constant Amplitude Removal Preprocessor (CARP)	33
4.1.4	Power Floor Preprocessor (PFP)	34
4.2	Speech features	46
4.2.1	Filter-bank spectral analysis	48
4.2.2	Formant frequencies- and bandwidths	50
4.2.3	Combining MFCCs and formant frequencies	51
4.2.4	Perceptual Linear Prediction (PLP) features	51
4.3	Feature normalisation	53
4.3.1	Feature scaling	53
4.3.2	Feature Mean Subtraction	53
4.3.3	Velocity and acceleration coefficients	53
4.3.4	Karhunen-Loeve Transform (KLT) [10, 22]	54
4.3.5	Feature Frame Expansion (FFE)	55
4.3.6	Feature Cross-term Expansion (FCE)	56
4.4	Vector Quantisation (VQ)	57
4.5	Speech modelling	59
4.5.1	Gaussian PDFs	59
4.5.2	Gaussian Mixture PDFs (GMMs)	60
4.5.3	Hidden Markov Models (HMMs)	60
4.5.4	Common SV Model (CSVm)	64
4.6	Hierarchical VQ (HVQ)	64
4.7	Speech classification	66
5	Recognition configurations for SAE SV classification	69
5.1	Applicable speech recognition configurations	69
5.1.1	Phoneme recognition followed by N-gram modelling	69
5.1.2	Gaussian Mixture Models (GMMs)	70
5.1.3	N-th order ergodic HMMs	71
5.1.4	Hierarchical HMMs (HHMMs)	73
5.2	Speech recognition configurations selected	73
6	Experimental investigation	76
6.1	Speech preprocessing experiment	79
6.2	Speech features experiment	82
6.3	Feature normalisation experiment	85
6.4	SV speech modelling experiments	88
6.4.1	Modelling SVs with GMMs	89

6.4.2	Initialisation of SV GMMs using a CSVM GMM	92
6.4.3	Investigating first-order HMM configurations for SV modelling . . .	94
6.4.4	Using GMMs as emitting densities in the SV HMMs	98
6.4.5	Investigating second-order HMM configurations for SV modelling .	101
6.4.6	Investigating a third-order HMM configuration for SV modelling . .	106
6.4.7	Using full covariance Gaussian PDFs as SV HMM emitting densities	108
6.5	Comparison of SV classification systems	110
6.5.1	Overall average RER attained by the SV classification systems . . .	112
6.5.2	Computational efficiency of the SV classification systems	116
7	Conclusion and recommendations	119
7.1	Key experimental results	119
7.2	Future work	120
7.3	Conclusion	121
A	AST SAE corpus	127
B	Detail experimental results	129
B.1	Computation of confidence intervals	129
B.2	Tables and graphs for experiments	130
4.3	Frame power p_{frame} for the PTF example utterance shown in Figure 4.1	
	Section (a) of this figure is the frame power for the complete utterance, whereas sections (b), (c) and (d) represent the frame power for the depending sections of Figure 4.1.	
4.4	Morphologically-filtered frame power p_{morph} for the PTF example utterance. An 'opening' filter was applied to the frame power shown in Figure 4.3 to remove noise. This was followed by application of a 'closing' filter to fill in potential silence sections. 'y1' and 'y2' are points in time which are utilised along with the threshold levels 'x1' and 'x2' to segment the utterance into speech and non-speech sections.	
4.5	Percentile-normalised frame power p_{pn} for PTF example utterance.	
4.6	Example utterance following PTF application.	

List of Figures

3.1	SAE corpus speech durations. Shown for each SV is the total duration speech available for the SV ('Total'), the total duration of the training utterances ('Training') and the total duration of the utterances used for testing ('Testing'). Also shown for each SV is the total duration of speech for which the caller gender and telephone type was unknown ('Unknown').	30
4.1	Example utterance used for PFP algorithm description. The complete utterance is shown in (a); with (b), (c) and (d) zooming in on selected sections of the example utterance. Sections (b) and (c) of the utterance contain speech whereas section (d) contains only recording artifacts.	38
4.2	Spectrogram for the PFP example utterance (shown in Figure 4.1). Section (a) is the spectrogram for the complete utterance shown in Figure 4.1 (a). Sections (b), (c) and (d) are the spectrogram sections for the corresponding sections of Figure 4.1.	39
4.3	Frame power \mathbf{p}_{frm} for the PFP example utterance (shown in Figure 4.1). Section (a) of this figure is the frame power for the complete utterance, whereas sections (b), (c) and (d) represent the frame power for the corresponding sections of Figure 4.1.	40
4.4	Morphologically-filtered frame power \mathbf{p}_{morph} for the PFP example utterance. An 'opening' filter was applied to the frame power shown in Figure 4.3 to remove spikes. This was followed by application of a 'closing' filter to fill in potential silence sections. 'p1' and 'p2' are percentile levels which are utilised along with the threshold levels 't1' and 't2' to segment the utterance into speech and non-speech sections.	41
4.5	Percentile-normalised frame power \mathbf{p}_{p2} for PFP example utterance.	43
4.6	Example utterance following PFP application.	43

4.7	Speech signal (a), spectrogram (b) and frame power \mathbf{p}_{frm} (c) for example utterance used to illustrate the removal of AGC contamination using the PFP. The AGC influence is visible in the divergence of the utterance signal amplitude in (a), between 3,5s and the end of the utterance. Correspondingly, the frame power waveform in (c) shows a sudden increase from the ambient power level at around 3,5s.	44
4.8	Morphologically filtered frame power \mathbf{p}_{morph} (a), percentile-normalised frame power \mathbf{p}_{p_2} (b) and modified speech signal (c) for the example utterance used to illustrate the removal of AGC contamination using the PFP.	45
4.9	Speech signal (a), spectrogram (b) and frame power \mathbf{p}_{frm} (c) for the example utterance used to illustrate the removal of call disconnect spikes using the PFP.	46
4.10	Morphologically filtered frame power \mathbf{p}_{morph} (a), percentile-normalised frame power \mathbf{p}_{p_2} (b) and modified speech signal (c) for the example utterance used to illustrate the removal of call disconnect spikes using the PFP.	47
4.11	Conceptual representation of filter-bank spectral analysis (after Arslan and Hansen).	49
4.12	Conceptual representation of the KLT.	56
4.13	Conceptual illustration of Feature Frame Expansion (FFE).	57
4.14	Initial clustering of an artificial feature space using binary split Vector Quantisation (VQ).	58
4.15	Final clustering for the artificial feature space of Figure 4.14, following application of the K-means algorithm.	59
4.16	Conceptual representation of a first-order HMM. Only two states of the HMM are shown.	61
4.17	Ergodic HMM initialised from a GMM. Transition weights for links terminating on a given HMM state are set equal to the weight of the corresponding GMM component weight. The corresponding GMM component density is utilised as HMM emitting density. The additional weight w_3 added to the HMM ensures that the HMM has a well-defined exit state.	63
4.18	Partial second-order HMM (after Du Preez and Weber [12]).	65
4.19	Third order context-emphasised left to right HMM with one state skip (after Du Preez and Weber [12]).	65
5.1	Applicable speech recognition configurations for SAE classification. Shown are the pattern modelling configurations available, possible methods of model initialisation (CSVM, GMM to HMM transform etc.) and whether a pattern modelling configuration is capable of using speech transcriptions. The lines in the figure connect these related concepts.	74

LIST OF FIGURES

x

6.1	Comparison of the average RER [%] attained with two different preprocessor configurations. The CARP configuration utilised a CARP preprocessor followed by preemphasis and power normalisation. The PFP configuration prepended a PFP preprocessor to the CARP configuration.	81
6.2	Average RER [%] for the speech feature extraction configurations investigated. MFCC9 and MFCC18 were nine- and eighteen-dimensional MFCCs respectively, computed from twenty-two filter bands over an analysis frequency range of 200Hz to 3.5kHz. FF consisted of formant frequencies computed from an eighth-order LP filter polynomial. MFCC9_FF consisted of the combination of the MFCC9 and FF feature extraction configurations. The AH configuration utilised the filter bank spectral analysis method of Arslan and Hansen. The PLP configuration consisted of eight-dimensional PLP cepstral coefficients with frame energy as additional feature.	84
6.3	Average RER [%] for the feature normalisation configurations investigated. Configuration NONE performed no feature normalisation. Configuration FS utilised Feature Scaling only. Configuration Δ _KLT used Δ (velocity) coefficients followed by KLT. Configuration $\Delta\Delta$ _KLT used $\Delta\Delta$ (acceleration) coefficients followed by KLT. Configuration KLT used only the KLT as normalisation. FFE_KLT utilised Feature Frame Expansion followed by KLT. FCE_KLT utilised Feature Cross-term Expansion followed by KLT.	87
6.4	Overall average RER [%] as a function of varying the component count for diagonal covariance GMMs.	90
6.5	Average RER [%] attained by seventy-component GMM- (GMM_70) and single Gaussian PDF (KLT) SV modelling.	91
6.6	Average RER [%] for 'standard' 70-component SV GMMs (GMM_70) vs. that attained by 70-component CSVM-initialised SV GMMs (GMM_CSVM).	93
6.7	Average RER [%] for the first-order HMM configurations investigated. Configuration X1 utilised conventional seventy-state ergodic HMMs with a single diagonal covariance Gaussian PDF per HMM state. Configuration X1_CSVM utilised a CSVM HMM to initialise the SV HMMs. Configuration X1_GMM initialised the SV HMMs using the GMMs of the GMM_70 configuration and the GMM to ergodic HMM transformation described in subsection 4.5.3. The X1_GMM_CSVM configuration utilised a CSVM HMM which was initialised from the CSVM GMM of the GMM_CSVM configuration.	96
6.8	Comparison of the average RER [%] attained by GMM modelling (GMM_CSVM configuration) and HMM modelling (X1_GMM_CSVM configuration) of the SVs.	98

6.9	Average RER [%] attained by HMM configuration which utilised GMMs as state emitting densities (X1_HVQ configuration) vs. that of the HMM configuration which utilised single Gaussian PDFs as state emitting densities (X1_CSVM configuration).	100
6.10	Average RER [%] for the second-order HMM configurations investigated. Configuration X2 converted the trained first-order SV HMMs from the X1_GMM_CSVM configuration to second-order models. Configuration X2_CSVM converted the trained CSVM HMM of the X1_GMM_CSVM configuration to a second-order CSVM HMM from which the SV HMMs were then initialised. The C2 configuration converted the trained first-order SV HMMs from the X1_GMM_CSVM configuration to second-order context-emphasised HMMs. The C2_CSVM configuration converted the trained CSVM HMM of the X1_GMM_CSVM configuration to a context-emphasised second-order CSVM HMM from which the SV HMMs were initialised. The D2 configuration converted the trained first-order SV HMMs from the X1_GMM_CSVM configuration to second-order duration-emphasised HMMs. The D2_CSVM configuration converted the trained CSVM HMM of the X1_GMM_CSVM configuration to a duration-emphasised second-order CSVM HMM from which the SV HMMs were initialised.	103
6.11	Average RER [%] attained by first-order HMM modelling (X1_GMM_CSVM configuration) and second-order HMM modelling (C2_CSVM configuration) of the SVs.	105
6.12	Average RER [%] attained by second-order HMM modelling (X2_CSVM configuration) and third-order HMM modelling (X3 configuration) of the SVs.	107
6.13	Comparison of the average RER [%] attained by the C2_CSVM HMM configuration (diagonal covariance Gaussian PDFs) and the C2_FC HMM configuration (full covariance Gaussian PDFs).	109
6.14	Average RER [%] for the major SV classification systems investigated. Also shown are the 95% confidence intervals (indicated by the error bars) for the average RER values, assuming independent Bernoulli trials.	114
B.1	CARP configuration RER [%].	131
B.2	PFP configuration RER [%].	131
B.3	MFCC9 configuration RER [%].	134
B.4	MFCC18 configuration RER [%].	134
B.5	FF configuration RER [%].	135
B.6	MFCC9_FF configuration RER [%].	135
B.7	AH configuration RER [%].	136

LIST OF FIGURES

xii

B.8 PLP configuration RER [%].	136
B.9 NONE configuration RER [%].	139
B.10 FS configuration RER [%].	140
B.11 Δ _KLT configuration RER [%].	140
B.12 $\Delta\Delta$ _KLT configuration RER [%].	141
B.13 KLT configuration RER [%].	141
B.14 FFE_KLT configuration RER [%].	142
B.15 FCE_KLT configuration RER [%].	142
B.16 GMM_70 configuration RER [%].	144
B.17 GMM_CSVM configuration RER [%].	144
B.18 X1 configuration RER [%].	146
B.19 X1_CSVM configuration RER [%].	147
B.20 X1_GMM configuration RER [%].	147
B.21 X1_GMM_CSVM configuration RER [%].	148
B.22 X1_HVQ configuration RER [%].	149
B.23 X2 configuration RER [%].	152
B.24 X2_CSVM configuration RER [%].	152
B.25 C2 configuration RER [%].	153
B.26 C2_CSVM configuration RER [%].	153
B.27 D2 configuration RER [%].	154
B.28 D2_CSVM configuration RER [%].	154
B.29 X3 configuration RER [%].	155
B.30 C2_FC configuration RER [%].	156

List of Tables

4.1	Centre frequencies [Hz] for filter-bank configuration of Arslan and Hansen.	50
4.2	Confusion matrix with corresponding classification accuracies and RER values for a typical classification experiment.	67
6.1	Test segment durations utilised in experiments.	77
6.2	Overall average RER [%] for the preprocessor configurations investigated. .	80
6.3	Overall average SV RER [%] for the speech feature extraction configurations investigated. MFCC9 and MFCC18 were nine- and eighteen-dimensional MFCCs respectively, computed from twenty-two filter bands over an analysis frequency range of 200Hz to 3.5kHz. FF consisted of formant frequencies computed from an eighth-order LP filter polynomial. MFCC9_FF consisted of the combination of the MFCC9 and FF feature extraction configurations. The AH configuration utilised the filter bank spectral analysis method of Arslan and Hansen. The PLP configuration consisted of eight-dimensional PLP cepstral coefficients with frame energy as additional feature.	83
6.4	Overall average RER [%] for the feature normalisation configurations investigated. Configuration NONE performed no feature normalisation. Configuration FS utilised Feature Scaling only. Configuration Δ _KLT used Δ (velocity) coefficients followed by KLT. Configuration $\Delta\Delta$ _KLT used $\Delta\Delta$ (acceleration) coefficients followed by KLT. Configuration KLT used only the KLT as normalisation. FFE_KLT utilised Feature Frame Expansion followed by KLT. FCE_KLT utilised Feature Cross-term Expansion followed by KLT.	88

6.5	Overall average RER [%] for the first-order HMM configurations investigated. Configuration X1 utilised conventional seventy-state ergodic HMMs with a single diagonal covariance Gaussian PDF per HMM state. Configuration X1_CSVM utilised a CSVM HMM to initialise the SV HMMs. Configuration X1_GMM initialised the SV HMMs using the GMMs of the GMM_70 configuration and the GMM to ergodic HMM transformation described in subsection 4.5.3. The X1_GMM_CSVM configuration utilised a CSVM HMM which was initialised from the CSVM GMM of the GMM_CSVM configuration.	97
6.6	Overall average RER [%] for the second-order HMM configurations investigated. Configuration X2 converted the trained first-order SV HMMs from the X1_GMM_CSVM configuration to second-order models. Configuration X2_CSVM converted the trained CSVM HMM of the X1_GMM_CSVM configuration to a second-order CSVM HMM from which the SV HMMs were then initialised. The C2 configuration converted the trained first-order SV HMMs from the X1_GMM_CSVM configuration to second-order context-emphasised HMMs. The C2_CSVM configuration converted the trained CSVM HMM of the X1_GMM_CSVM configuration to a context-emphasised second-order CSVM HMM from which the SV HMMs were initialised. The D2 configuration converted the trained first-order SV HMMs from the X1_GMM_CSVM configuration to second-order duration-emphasised HMMs. The D2_CSVM configuration converted the trained CSVM HMM of the X1_GMM_CSVM configuration to a duration-emphasised second-order CSVM HMM from which the SV HMMs were initialised. . . .	104
6.7	Overall average RER [%] for the major SV classification systems investigated as well as the average RER attained on each test segment duration. Also shown are the 95% confidence intervals (in square brackets) for the average RER values, assuming independent Bernoulli trials.	113
6.8	Confusion matrix for X1_GMM_CSVM configuration on 30s test segments.	115
6.9	Confusion matrix for X1_GMM_CSVM configuration on 120s test segments.	115
6.10	Confusion matrix for C2_FC configuration on 30s test segments.	116
6.11	Confusion matrix for C2_FC configuration on 120s test segments.	116
6.12	Processing delay for the major SV classification systems investigated. . . .	117
A.1	SAE corpus speech statistics.	127
A.2	SAE corpus speech statistics following application of PFP preprocessor. . .	128
B.1	CARP configuration RER [%].	130
B.2	PFP configuration RER [%].	130

B.3 MFCC9 configuration RER [%].	132
B.4 MFCC18 configuration RER [%].	132
B.5 FF configuration RER [%].	132
B.6 MFCC9_FF configuration RER [%].	133
B.7 AH configuration RER [%].	133
B.8 PLP configuration RER [%].	133
B.9 NONE configuration RER [%].	137
B.10 FS configuration RER [%].	137
B.11 Δ _KLT configuration RER [%].	137
B.12 $\Delta\Delta$ _KLT configuration RER [%].	138
B.13 KLT configuration RER [%].	138
B.14 FFE_KLT configuration RER [%].	138
B.15 FCE_KLT configuration RER [%].	139
B.16 GMM_70 configuration RER [%].	143
B.17 GMM_CSVM configuration RER [%].	143
B.18 X1 configuration RER [%].	145
B.19 X1_CSVM configuration RER [%].	145
B.20 X1_GMM configuration RER [%].	145
B.21 X1_GMM_CSVM configuration RER [%].	146
B.22 X1_HVQ configuration RER [%].	148
B.23 X2 configuration RER [%].	149
B.24 X2_CSVM configuration RER [%].	150
B.25 C2 configuration RER [%].	150
B.26 C2_CSVM configuration RER [%].	150
B.27 D2 configuration RER [%].	151
B.28 D2_CSVM configuration RER [%].	151
B.29 X3 configuration RER [%].	151
B.30 C2_FC configuration RER [%].	155

* FFE

* FCE

* FMS

* GMM

* HGMM

* ITSM

* HVQ

List of acronyms

- AE Afrikaans English (section 3.1).
- AGC Automatic Gain Control (subsection 4.1.4).
- AST African Speech Technology (subsection 1.1.3).
- BE Black English (section 3.1).
- CARP Constant Amplitude Removal Preprocessing (subsection 4.1.3).
- CE Coloured English (section 3.1).
- CSVM Common Spoken Variant Model (subsection 4.5.4).
- DCT Discrete Cosine Transform (subsection 4.2.1).
- DFT Discrete Fourier Transform ([33]).
- DSP Digital Signal Processing.
- EE English English (section 3.1).
- EM Expectation Maximisation algorithm (5.1.2).
- FCE Feature Cross-term Expansion (subsection 4.3.6).
- FFE Feature Frame Expansion (subsection 4.3.5).
- FFT Fast Fourier Transform[33].
- FMS Feature Mean Subtraction (subsection 4.3.2).
- GMM Gaussian Mixture Model (subsection 4.5.2).
- HHMM Hierarchical Hidden Markov Model (section 5.1).
- HMM Hidden Markov Model (subsection 4.5.3).
- HVQ Hierarchical Vector Quantisation (section 4.6).

LIST OF ACRONYMS

xvii

- IE Indian English (section 3.1).
- ISDN Integrated Services Digital Network (section 3.2).
- KLT Karhunen-Loeve Transform (subsection 4.3.4).
- LC Language Category (chapter 2).
- LDA Linear Discriminant Analysis (subsection 6.4.7).
- LID Language Identification (chapter 2).
- LP Linear Prediction (subsection 4.2.2).
- LVCSR Large Vocabulary Continuous Speech Recognition (2.2.2).
- MCU Mixture Component Usage (2.2.2).
- MFCC Mel Frequency Cepstral Coefficient (subsection 4.2.1).
- ML Maximum Likelihood (section 4.7).
- MLE Maximum Likelihood Estimation (subsection 4.5.1).
- MSE Mean Square Error (section 4.4).
- NIST American National Institute of Standards and Technology (section 1.3).
- OGI-TS Oregon Graduate Institute Telephone Speech corpus[30].
- PCA Principal Component Analysis (subsection 4.3.4).
- PFP Power Floor Preprocessing (subsection 4.1.4).
- PPR Parallel Phoneme Recognition (2.2.2).
- PRLM Phone Recognition followed by Language Modelling (2.2.2).
- PRLM-P Parallel Phone Recognition followed by Language Modelling (2.2.2).
- RER Recognition Error Rate (section 4.7).
- SED Speech Endpoint Detection (subsection 4.1.4).
- SAE South African English (subsection 1.1.3).
- SNR Signal to Noise Ratio (subsection 4.1.4).
- SV Spoken Variant (subsection 1.1.2).

LIST OF ACRONYMS

xviii

- UBM Universal Background Model ([36]).
- VPF Vector of Phoneme Frequencies (subsection 2.2.2).

List of symbols

- μ_p Preemphasis coefficient (Equation 4.1).
- win_{dur} CARP analysis window length (subsection 4.1.3).
- ret_{dur} CARP/PFP retention duration (subsection 4.1.3,subsection 4.1.4).
- \mathbf{v} Speech vector (Equation 4.2).
- \mathbf{p}_{frm} Frame power vector (Equation 4.2).
- frm_{dur} PFP internal frame length (subsection 4.1.4).
- frm_{ovl} PFP internal frame skip (subsection 4.1.4).
- $peak_{min}$ PFP minimum peak duration (subsection 4.1.4).
- gap_{min} PFP minimum gap duration (subsection 4.1.4).
- \mathbf{p}_{morph} Morphologically filtered frame power vector (subsection 4.1.4).
- p_1 PFP first percentile (subsection 4.1.4).
- t_1 PFP first threshold (Equation 4.3).
- o_1 PFP first offset (Equation 4.3).
- \mathbf{p}_{p_1} PFP first percentile-normalised frame power (Equation 4.4).
- p_2 PFP second percentile (subsection 4.1.4).
- t_2 PFP second threshold (Equation 4.5).
- o_2 PFP second offset (Equation 4.3).
- \mathbf{p}_{p_2} PFP second percentile-normalised frame power (Equation 4.6).
- $c(n)$ N'th cepstral coefficient (Equation 4.7).
- $S(m)$ Log-energy output of m'th filter band (Equation 4.8).

- X DFT of speech signal (Equation 4.8).
- H_m Windowing function for m 'th filter band (Equation 4.8).
- f_{mel} Frequency measured according to Mel scale (Equation 4.9).
- f Frequency measured in Hz (Equation 4.9).
- T Sampling period (Equation 4.11).
- f_i i 'th LP filter polynomial root formant frequency (Equation 4.11).
- b_i i 'th LP filter polynomial root formant bandwidth (Equation 4.13).
- f_{bark} Frequency measured according to Bark scale (Equation 4.14).
- \mathbf{x}_i Feature matrix column i (Equation 4.18).
- $\bar{\mathbf{x}}_i$ Mean of i 'th feature matrix column (Equation 4.18).
- $\hat{\mathbf{x}}_i$ Modified feature matrix column i (Equation 4.18).
- \mathbf{X} Represents a feature matrix.
- $\hat{\mathbf{S}}$ Feature scaling diagonal matrix (Equation 4.17).
- $\hat{\mathbf{X}}$ Modified feature matrix (Equation 4.17).
- $\mathbf{x}_i(j)$ Feature matrix column i , element j (Equation 4.19).
- $\Delta \mathbf{x}_i(j)$ Δ coefficient for feature matrix column i , element j (Equation 4.19).
- \mathbf{P}^T PCA transformation matrix (Equation 4.21).
- Σ Covariance matrix prior to application of \mathbf{P}^T (Equation 4.22).
- \mathbf{D} Covariance matrix following application of \mathbf{P}^T (Equation 4.22).
- $\%_{ret}$ PCA information retention percentage (subsection 4.3.4).
- α_k k 'th retained eigenvalue used in PCA (subsection 4.3.4).
- $trace(\mathbf{D})$ Sum of diagonal elements (eigenvalues) in \mathbf{D} (subsection 4.3.4).
- $d(\mathbf{x}, \mathbf{y})$ VQ quantisation error (Equation 4.25).
- $N(\bar{\mathbf{x}}, \mathbf{C})$ Gaussian PDF with centroid vector $\bar{\mathbf{x}}$ and covariance matrix \mathbf{C} (subsection 4.5.1).
- ρ Correlation coefficient (subsection 4.5.1).
- C_{ij} Covariance for two random variables i and j (subsection 4.5.1).

Chapter 1

Introduction

This thesis presents the results of research conducted on the automatic classification of South African English (SAE) Spoken Variants (SVs), using a transcription-less speech recognition approach.

1.1 Motivation

1.1.1 Multi-linguality

The ability to process more than one language is an important consideration for the development of any speech recognition application in South Africa. This is because South Africa has eleven official languages [2]. In order to enable as many speakers as possible to access a speech recognition application using their first language, the application would ideally support all eleven official South African languages.

Financial constraints might prevent the multi-linguality ideal from being realised. Gathering speech for the development of a speech recognition application is a labour- and time intensive process and therefore it is expensive. The costs involved are proportional to the number of languages supported by the application. Hence it will be expensive to develop a speech recognition application using all eleven official languages simultaneously.

It might be possible to utilise only one of the eleven official languages in preliminary development of the speech recognition application, depending on the aspect(s) of speech recognition addressed by the particular application. This should reduce initial system development costs significantly.

1.1.2 Spoken Variants (SVs)

For any South African official language one wishes to recognise, it would seem logical to develop the ability to recognise the language in all its spoken versions. These different spoken versions or variants of a language are known either as accents or dialects of the language, according to a number of linguistic criteria.

If one has a speech recognition system which is capable of recognising many of the accents or dialects of a given language, this system will be more robust to accent and dialect variations which may otherwise detract from the system's performance. We will next present a brief description of accent and dialect, after which we will motivate the unification of these linguistic concepts from a pattern recognition viewpoint.

Accents [8]

There is an audible difference between the speech of a first language speaker of a language and that of someone who is not a first language speaker of the given language. A first language speaker of a language is sometimes called a 'native' speaker of the language, whereas someone who is not a first language speaker of the language is called a 'non-native' speaker of the language.

The audible difference in the speech of native and non-native speakers of a language results from the different spectral- and temporal characteristics of their pronunciations. The speech of a non-native speaker of a language is called 'accented' speech or the speaker is said to 'speak with an accent'.

Dialects [8, 27]

Dialects are different sounding versions of the same language (base language) as spoken by different groups of native speakers. Where accents differ only in terms of their pronunciation, dialects often differ in terms of their grammar and vocabulary as well. The speaker groups representing different dialects of a base language tend to differ in terms of their geographical location and their social grouping.

Unifying the concepts of accent and dialect

The distinction between accent and dialect is not always based on linguistic criteria alone. The linguistic distinction appears to be influenced by historical, political and social motivations [8].

Both accents and dialects are primarily the result of pronunciation differences between

multiple speaker groups speaking the same base language. These pronunciation differences result in different spectral and temporal characteristics for the speech of the different speaker groups. As far as pattern recognition is concerned, both accents and dialects will be modelled and recognised by encoding their spectral and temporal characteristics with the appropriate pattern modelling techniques.

From a pattern recognition viewpoint, accents and dialects may then simply be seen as different-sounding versions or variants of the same base language which are modelled in terms of their temporal and spectral characteristics. These variants will be known as Spoken Variants (SVs) throughout this thesis.

Applications utilising SV classification

There are a number of possible applications for a speech recognition system which is capable of classifying the SVs of a language. Such a system could be used to determine the first language of users (accent), based on the SV classification of their speech. The system could also be used to determine the probable geographical region of origin or social group (dialect) of the users, based on the SV classification of their speech.

This capability might be useful in a system where initial speech input to the system is constrained to one language. If SV classification of a user's speech determines a first-language mismatch between the user and the system, the system might switch to a language model which matches the user's first language. This model could be used by the system both for presenting speech prompts to the user and for processing speech input from the user.

Another use for SV classification of speech is in criminal forensic profiling of speakers. SV classification could be used to determine the first language, possible geographical location of residence and even social group of criminals from a recording of their speech. The use of speech recognition for criminal forensic applications is briefly discussed in [8].

It is hypothesised that techniques developed for classifying the SVs of one language should be applicable for classifying the SVs of another; perhaps with minor modification.

1.1.3 South African English (SAE)

This thesis used South African English (SAE) for the investigation of SV classification techniques. SAE was chosen for investigation, simply because there was a speech corpus available for it when the thesis experimental investigation commenced. The SAE speech

used in this thesis was collected as part of the African Speech Technology (AST) project [1].

The AST project was probably the first of its kind in South Africa where large speech corpora were created for some of the South African official languages. The main aim of the AST project was to develop an automated hotel telephone booking system in five of the official South African languages. This thesis was conducted separately from the development of the telephone booking system.

SAE was one of the official languages for which speech was collected as part of the AST project. For the AST SAE corpus, five SVs were identified. The AST SAE corpus and its SVs are described in section 3.1.

1.1.4 Speech transcriptions

For many of the modern speech corpora, text transcriptions are produced for the corpus speech. The speech is typically transcribed at the phoneme level and often the time-alignment of the phonemes are provided. Speech transcriptions provide additional information which is not present in a system which utilises only spectral features.

Perhaps the most important use of speech transcriptions is that they provide a set of concrete speech units, such as phonemes, in terms of which one may model the categories which are to be recognised by the system. By modelling phoneme interactions, such as with word models, one may impose higher-level linguistic constraints which can improve the performance of a speech recognition system.

Producing accurate speech transcriptions for any speech corpus is probably more labour-intensive than the gathering of the speech itself. This is because the gathering of speech for many modern speech corpora is largely an automated process, whereas the transcription of speech often requires the involvement of humans trained specifically for this purpose.

Since the speech transcription process is labour-intensive, it will be expensive. The financial costs involved in producing speech transcriptions might force developers of a speech recognition system to utilise a transcription-less speech recognition approach. Speech recognition systems which do not make use of speech transcriptions also tend to be simpler than systems which do utilise speech transcriptions.

A transcription-less speech recognition approach is ideal for initial system development because of the reduced development costs and possible reduction in system complexity it

affords. Speech transcriptions were produced for the AST SAE corpus, but the transcription process was not finalised when the thesis implementation stage commenced. This excluded the use of speech transcriptions in the development of the SAE SV classification system.

1.2 Objectives

- Create a pattern recognition system capable of automatic classification of SAE SVs.
- The speech recognition system must operate without speech transcriptions.

1.3 Contributions

- Developed a pattern recognition system which is capable of classifying SAE SVs without requiring speech transcriptions (section 6.5).
- Showed that it is possible to systematically decrease the system overall average Recognition Error Rate (RER, see chapter 6) from 52,05% to 20,03% (Figure 6.14).
- Showed a reduction in overall average RER as test segment durations were increased (subsection 6.5.1). This result has positive implications for criminal forensics applications where one might encounter speech segment durations in excess of one minute.
- Implemented a speech preprocessor which removes portions of input speech which have no amplitude variation. This preprocessor, the Constant Amplitude Removal Preprocessor (CARP), was utilised to remove empty speech files automatically and to prevent numerical instabilities during feature extraction (subsection 4.1.3).
- Implemented a speech preprocessor which improves system robustness against non-speech artifacts (subsection 4.1.4). This preprocessor, the Power Floor Preprocessor (PFP), was designed in collaboration with the thesis supervisor, Prof. J.A. du Preez. Experimental results confirmed that this preprocessor improves classification performance on the AST SAE corpus (section 6.1).
- Showed that Perceptual Linear Prediction (PLP, subsection 4.2.4) cepstra outperform Mel Frequency Cepstral Coefficients (MFCCs, subsection 4.2.1) on the SAE SV classification task (Figure 6.2).
- Showed that use of a global initialisation model, the CSVM (Common SV Model, subsection 4.5.4) improves overall SV classification performance (subsection 6.4.2, subsection 6.4.3 and subsection 6.4.5).

- Showed that Hidden Markov Models (HMMs, subsection 4.5.3) outperform Gaussian Mixture Models (GMMs, subsection 4.5.2) on the SAE SV classification task (Figure 6.8 and Table 6.7).
- Showed that a reduction in overall average RER is attained when diagonal covariance Gaussian emitting densities of the HMMs are replaced with full covariance Gaussian emitting densities (Figure 6.13).
- Highlighted the trade-off between classification accuracy and computational efficiency (subsection 6.5.1 and subsection 6.5.2).
- Implemented a Hierarchical Vector Quantisation (HVQ) process which allows one to simultaneously initialise the component centroids of multiple mixture probability density functions, directly from the spectral feature space (section 4.6). HVQ was utilised to investigate the use of GMMs as HMM emitting densities (subsection 6.4.4).
- Implemented the Mel Frequency Cepstral Coefficient (MFCC) feature representation (subsection 4.2.1) in software. This was required for investigating MFCCs as feature representation.
- Software implementation of filter-bank spectral analysis. This was required for investigation of the feature representation proposed by Arslan and Hansen (subsection 4.2.1). The filter-bank spectral analysis was implemented in a manner which makes it possible to use filter bands which are spaced according to an arbitrary frequency scale, including the Mel frequency scale (for the computation of MFCCs).
- Implemented a formant frequency estimator for investigating formant frequencies as speech features (subsection 4.2.2).

1.4 Overview

A literature study was conducted before the thesis experimental investigation commenced. The purpose of the literature study was to determine the state of existing speech recognition research in areas related to this thesis. The literature study is presented in chapter 2. The speech recognition areas of language identification, accent recognition and dialect recognition were deemed to be of interest to this thesis.

In order to make the discussion more general, languages and SVs were grouped together under the label LC (language category) for the literature study. The trend which emerged

from the literature study is that LCs are modelled using one of two generic speech recognition configurations. The first configuration uses a combination of phoneme recognisers and N-gram language models trained from speech transcriptions whereas the second configuration uses GMMs or ergodic HMMs which are trained without speech transcriptions (section 2.3).

In chapter 3 the AST SAE corpus is described. The SVs which comprise the AST SAE corpus; namely Afrikaans English (AE), Black English (BE), Coloured English (CE), English English (EE) and Indian English (IE) are described in section 3.1. The corpus speech collection process and problems encountered with the corpus speech are briefly described in section 3.2. These problems included speech files which contained no speech at all as well as non-speech artifacts introduced by the speech transmission channel and recording hardware.

The duration of speech available on a per-SV basis is shown in section 3.3. The speech duration figures reveal that for more than 78% of the total duration AST SAE speech, the speaker gender and telephone type (land line vs. cell phone) was not known. This prevented the development of gender-specific as well as telephone-specific speech models, which could improve the classification performance of the system significantly.

The theoretical concepts utilised in this thesis are presented in chapter 4. Briefly discussed in this chapter are the theoretical concepts from pattern recognition which are utilised in this thesis. Specifically of interest in this chapter are the following speech processing- and pattern recognition techniques implemented for this thesis:

- CARP (Constant Amplitude Removal Preprocessor). This preprocessor was designed to remove constant amplitude speech sections which cause numerical instabilities during feature extraction (subsection 4.1.3).
- PFP (Power Floor Preprocessor, subsection 4.1.4). This preprocessor helped improve system robustness against the non-speech artifacts present in the AST SAE corpus (section 6.1).
- HVQ (Hierarchical Vector Quantisation, section 4.6). HVQ may be used to initialise multiple GMMs from the spectral feature space without requiring speech transcriptions. HVQ was utilised in the experiment which investigated the use of GMMs as HMM emitting densities (subsection 6.4.4).
- CSVM (Common SV Model, subsection 4.5.4), which is used to provide better parameter initialisation of SV models. This helps to alleviate the local optima problem

encountered with model parameter estimation through the Expectation Maximisation (EM) algorithm. Use of the CSVM improved the overall SV classification performance (subsection 6.4.2, subsection 6.4.3 and subsection 6.4.5).

A number of speech recognition configurations considered for the classification of SAE SVs are presented in chapter 5. These configurations were variations on the two general speech recognition configurations identified in the literature synopsis of chapter 2. The theoretical basis for these configurations is provided in chapter 4.

The thesis objectives state that the speech recognition system used for classification of the SAE SVs has to operate without the use of speech transcriptions, therefore configurations which require speech transcriptions were not considered for implementation. It was decided to proceed with the implementation of speech recognition configurations utilising GMMs and ergodic HMMs trained on spectral features only (section 5.2).

In chapter 6 the thesis experimental investigation is presented. The experimental investigation consisted of a number of experiments which investigated each of the processing stages of a typical speech recognition system and the effect on the system overall classification performance when using a number of different configurations for each of these stages.

The experiments utilised seven different test segment lengths ranging from two seconds to five minutes in duration. These test segments were presented to the SV models in parallel and the SV model which yielded the highest score on a given test segment was deemed to be the classified category.

The experiments utilised the overall average Recognition Error Rate (RER) of the SV classification systems as comparison measure. The overall average RER was taken as the average RER attained over all the test segment lengths. The configuration for a given stage of the classification system which minimised the system overall average RER was retained in subsequent experiments. The classification systems which were created as a result of using different configurations for a given stage of the speech recognition system were labelled for ease of reference.

A number of experiments were conducted using very simple pattern modelling (single Gaussian PDFs) of the SAE SVs in order to determine the speech preprocessing-, speech feature extraction- and feature normalisation configurations to use for the remainder of the experiments (section 6.1,(section 6.2) and (section 6.3)).

The speech preprocessor experiment revealed that addition of the PFP preprocessor (subsection 4.1.4) to an existing speech preprocessing configuration resulted in a reduction of 7,78% for the overall average RER, from 56,44% to 52,05% (section 6.1).

The speech feature extraction experiment showed that using PLP (Perceptual Linear Prediction) features instead of the MFCCs used in the preprocessor experiment; an 11,9% reduction in the overall average RER, from 52,05% to 45,86%, could be attained (section 6.2).

An additional decrease of 6,39% in the overall average RER (from 45,86% to 42,93%) was attained when the Karhunen-Loeve Transform (KLT, subsection 4.3.4); a synonym for Principal Component Analysis (PCA), was used as feature normaliser instead of the feature scaling normaliser used in the previous experiments (section 6.3).

Next, a number of experiments were conducted to investigate the use of different pattern modelling techniques for modelling the SVs. First a number of experiments involving Gaussian Mixture Models (GMMs) were conducted. The first of these experiments attempted to establish the number of mixture components to use for the GMMs, based on the overall average RER obtained with different mixture component counts. The GMM mixture component counts were varied between two and hundred-and-twenty-eight, in steps of two (subsection 6.4.1).

It was determined that a mixture component count of seventy yielded the lowest RER of the component counts investigated. Using a seventy-component GMM instead of the single Gaussian PDF used in previous experiments, reduced the overall average RER by 3,14% from 42,93% to 41,58%.

The next experiment investigated the use of a CSVM (Common Spoken Variant Model) GMM to initialise the SV GMMs with. This CSVM GMM was trained on the acoustic features of all the SVs and was used to improve the initial conditions for the SV GMMs prior to parameter estimation with the EM algorithm. Use of the CSVM GMM to initialise the SV GMMs resulted in a RER decrease of 14,77% from 41,58% to 35,44% (subsection 6.4.2).

A number of experiments were conducted to investigate different first-order ergodic HMM configurations for modelling of the SVs. To facilitate direct comparison of results attained with the GMMs used in previous experiments, the first-order ergodic HMMs used a state count of seventy.

The first of these experiments investigated different methods of initialising first-order ergodic HMMs (subsection 6.4.3). The purpose of the parameter initialisation experiments were to determine the influence of HMM parameter initialisation on the overall classification performance of the system. Using a CSVM GMM to initialise a CSVM ergodic HMM; from which the SV HMMs were initialised, the overall average RER attained by the seventy-component GMM configuration was reduced by 32,42%, from 35,44% to 23,95%.

The CSVM HMM was initialised via a transformation which converts a GMM to a first-order ergodic HMM which is equivalent to the GMM as far as application of the EM algorithm is concerned. This transformation is described in subsection 4.5.3. The improvement in RER obtained by this configuration was attributed to better initialisation of the HMM acoustic component by using a CSVM GMM as seed for the CSVM HMM.

The experiment presented in subsection 6.4.4 investigated the use of GMMs as HMM emitting densities. This experiment utilised the Hierarchical Vector Quantisation (HVQ) process to create the VQ codebooks from which the centroids of the GMM state emitting densities were initialised. HVQ (section 4.6) was specifically implemented for this experiment.

The CSVM HMM which was initialised from a CSVM GMM could not be utilised as the basis for investigating the use of GMMs as emitting densities in the SV HMMs. This is because the emitting densities of this HMM would always consist of the single component densities of the original GMM CSVM source, by nature of the GMM to HMM transformation (subsection 4.5.3) utilised to create the CSVM HMM.

The second-best first order HMM configuration was therefore used instead. This configuration was easily modified to utilise GMMs instead of single Gaussian PDFs as HMM emitting densities. The experimental results showed that use of GMMs as HMM emitting densities resulted in a 2,53% improvement in overall average RER from the 24,9% of the configuration used as basis, to the eventual 24,27% of the configuration utilising GMM emitting densities (subsection 6.4.4). This improvement in overall average RER was shown not to be statistically significant.

Following the first-order HMM experiments, a number of second-order HMM configurations were investigated. The purpose of these experiments were to determine if the additional modelling complexity provided by higher-order HMM configurations could improve the overall average RER of the system. All of the second-order HMM configurations

utilised the best-performing first-order HMM configuration as basis.

The best-performing second-order HMM configuration converted the CSVM HMM of the best-performing first-order HMM configuration to a second-order context-emphasised HMM [11]. This converted CSVM HMM was retrained, the SV HMMs initialised from the trained CSVM HMM and then the SV HMMs were retrained prior to classification. This second-order HMM configuration reduced the overall average RER obtained by the best-performing first-order HMM configuration by 4,22% from 23,95% to 22,94% (subsection 6.4.5).

The best-performing second-order HMM configuration was used as starting point for a third-order HMM configuration. Unfortunately the resultant HMM had 677 798 states with 1 849 387 links, which made it impractical to train with the computer hardware at our disposal.

The second-best second-order HMM configuration was therefore used as starting point instead. This second-order configuration utilised standard second-order HMMs which typically contain far fewer links than the second-order context-emphasised HMMs. The third-order HMM configuration thus created did not attain a lower overall average RER than the best second-order HMM configuration.

The final experiment for the thesis is presented in subsection 6.4.7. In this experiment we investigated the effect of replacing the diagonal covariance Gaussian PDF HMM emitting densities of the best-performing second-order HMM configuration with full covariance Gaussian PDFs.

Diagonal covariance Gaussian PDFs were used in the majority of the thesis experiments; as GMM component densities as well as HMM emitting densities. This is because diagonal covariance Gaussian PDFs have fewer parameters than full covariance Gaussian PDFs, which make them computationally more efficient.

The computational efficiency of diagonal covariance Gaussian PDFs could be offset by a reduction in overall classification performance; depending on the nature of the feature space being modelled. When using diagonal covariance Gaussian PDFs, one assumes that the feature space dimensions are uncorrelated. The experiment presented in subsection 6.4.7 investigated this assumption.

The second-order HMM configuration utilising full covariance Gaussian PDF emitting

densities improved the overall average RER of the previously best-performing second-order HMM configuration, which utilised diagonal covariance Gaussian PDF emitting densities, by 12,68%, from 22,94% to 20,03%. This improvement in overall average RER was attained at a significant increase in computational cost. The duration of the classification process for the configuration utilising full covariance Gaussian PDF emitting densities was 22% higher than that of the configuration utilising diagonal covariance Gaussian PDF emitting densities.

In section 6.5 the SV classification configurations developed during the thesis experimental investigation are compared with each other. The configurations are compared in terms of the overall average RER attained by each, as well as the computational cost of each. The computational cost for a configuration is measured in terms of the duration of the feature extraction process as well as the duration of the classification process for the configuration. The reasons for not utilising the duration of the model parameter estimation process as part of the cost assessment are presented in subsection 6.5.2.

A review of the results of the experimental investigation will reveal that the thesis objectives were met. That is, we succeeded in developing an SV classification system which does not require speech transcriptions; within the thesis constraints. The shortcomings of the eventual SV classification system developed are that its overall average RER drops below 10% only once the test segment durations are increased above 60s and it has a high computational cost. This means that it cannot be deployed in real-time speech recognition applications.

These shortcomings are addressed in subsection 6.5.2 and the thesis conclusion chapter (chapter 7).

Chapter summary

In this chapter we presented the multi-linguality requirement as motivation for the thesis (subsection 1.1.1). This was followed by a description of the Spoken Variant (SV) concept. The SV concept unifies the linguistic concepts of accent and dialect, since these concepts are very similar from a pattern recognition viewpoint (subsection 1.1.2).

The reasons for using the AST SAE speech corpus to investigate SV classification was presented in subsection 1.1.3. The requirement for developing a transcription-less SV classification system was presented in subsection 1.1.4, which was followed by the thesis objectives in section 1.2.

The thesis contributions were presented in section 1.2, after which an overview of the thesis work; including the key experimental results, was presented in section 1.4. The following chapter presents the thesis literature study, in which we investigate previous research conducted in speech recognition fields related to this thesis.

Literature synopsis

Introduction

This chapter reviews previous work conducted in terms of speech recognition systems which were deemed relevant to this thesis. No literature describing research in speech recognition involving SAE SVs was found. Therefore the literature study focuses on existing work involving the SV subcomponents of accent and dialect, as well as the more general problem of language identification (LID).

The majority of papers reviewed here and deemed to be relevant to the SAE SV identification problem were from the LID field. Throughout this chapter, where the discussion is general enough such that one could refer to SVs or languages, the term ‘language categories’ (LCs) will be used instead. The following will be discussed in this chapter:

- The feature representations used for LC recognition (section 2.1).
- The speech recognition techniques employed for LC recognition and the results obtained (section 2.2).
- Conclusions drawn from the literature synopsis (section 2.3).

2.1 Feature representations used for LC recognition

2.1.1 Spectral speech features

Typical speech recognition systems utilise the speech spectral characteristics as features. Feature representations which parameterise speech spectral characteristics in one form or the other include MFCCs [7, 40, 16], LPC (Linear Prediction Coefficients [12]) equivalents, PLP (Perceptual Linear Prediction [15]) features. Formant frequencies and their band widths are another set of spectral features which could be utilised for speech recognition.

Chapter 2

Literature synopsis

Introduction

This chapter reviews previous work conducted in areas of speech recognition research which were deemed relevant to this thesis. No literature describing previous work on speech recognition involving SAE SVs was found. Therefore the literature study focused on existing work involving the SV subcomponents of accent and dialect, as well as the more general problem of language identification (LID).

The majority of papers reviewed here and deemed to be relevant to the SAE SV classification problem were from the LID field. Throughout this chapter, where the discussion is general enough such that one could refer to SVs or languages, the term ‘language categories’ (LCs) will be used instead. The following will be discussed in this chapter:

- The feature representations used for LC recognition (section 2.1).
- The speech recognition techniques employed for LC recognition and the results obtained (section 2.2).
- Conclusions drawn from the literature synopsis (section 2.3).

2.1 Feature representations used for LC recognition

2.1.1 Spectral speech features

Typical speech recognition systems utilise the speech spectral characteristics as features. Feature representations which parameterise speech spectral characteristics in one form or the other include MFCCs [7, 40, 16], LPC (Linear Prediction Coefficients [12]) cepstra or PLP (Perceptual Linear Prediction [15]) features. Formant frequencies and their bandwidths are another set of spectral features which could be utilised for speech recognition

[21]. Even if ‘higher-level’ features such as units of speech (phonemes, words etc.) are to be used for LC recognition, these features are parameterised in terms of their ‘lower-level’ spectral features.

2.1.2 Units of speech (phoneme, word etc.)

Different units of speech such as words or phonemes may be used to distinguish between LCs. The distinction may be represented by the difference in the inventory of speech units (broadly speaking, ‘vocabulary’) of the LCs. If the same speech unit inventory is used for the LCs, the LCs may be classified in terms of the differences in their spectral and temporal realisation of the same speech units.

Additionally, the temporal realisation of speech units may also be used to classify LCs. From the literature reviewed, it appears as if phonemes are the speech units of choice for the majority of language identification (LID) systems [35, 25].

2.1.3 Prosodic features

Prosodic features include the following [18]:

- The pauses which demarcate phrases.
- Pitch (fundamental speech frequency or F0).
- Rate (timing or rhythm) of phonemes.
- Volume of speech (energy).

Pitch and frame energy prosodic features (as well as their first- and second-order derivatives) were used by Fung and Kat for accent identification [21]. More detail on the work of Fung and Kat is presented in subsection 2.2.2.

2.2 Speech recognition techniques employed for LC recognition

2.2.1 LC recognition utilising spectral speech features

Language identification

LID using Gaussian Mixture Models (GMMs) and Hidden Markov Models (HMMs) trained on the general spectral characteristics of the target languages was presented by Zissman in [40, 41]. The methods discussed in [40] form the basis for the experimental

investigation of this thesis. This appears to be one of the first papers by Zissman on the LID problem. Each of the languages to be recognised were represented by continuous density, ergodic HMMs. For each language's HMM, tied-mixture GMMs (TGMMs) were used as the output probability densities.

The GMM centroids were initialised prior to training using binary-split VQ followed by refinement with the K-means algorithm. Each language had two separate TGMMs. The TGMMs were trained on separate parallel feature vector streams. The first feature vector stream consisted of MFCCs of unmentioned dimension, with the second feature vector stream consisting of the first order derivatives (delta coefficients) for the first feature vector stream.

Between one and twenty states were used for each language's HMM, with the TGMMs consisting of between four and hundred mixture components. Four different speech corpora were used for the LID experiments. Of these four corpora, only the OGI-TS corpus appears in later LID work. Three classification experiments were conducted on ten languages from the OGI-TS corpus.

The first experiment consisted of nine two-language classification tests involving English and the other nine languages in turn. The second experiment consisted of ten one-language detection experiments, i.e. language N vs. the rest. The final test consisted of one ten-way LID experiment. For this thesis the ten-way LID experiment would have been the most interesting, but unfortunately this experiment was selected for testing a single-state HMM, which is equivalent to a GMM.

The GMM modelling for experiment three actually utilised a single-state HMM per model, with the remaining languages pooled together to form a background GMM model. It is unclear from the paper whether the test utterance duration for the OGI-TS experiments was also set at 10s as with the other three corpora used in this paper. For the two-language classification task, Zissman obtained an average recognition error rate (RER) of 20%. The one-language detection experiment yielded an average RER of 27%. The ten-way LID experiment had a RER of 54%.

Du Preez and Weber applied higher-order HMMs to pair-wise language recognition on the OGI-TS corpus [11]. The purpose of the work presented in this paper was to determine the applicability of high-order HMM speech modelling techniques [12] to the language recognition problem. The authors utilised LPC-cepstra and their derivatives as features and did not use speech transcriptions at all.

A number of different HMM configurations were investigated, including first-order HMMs (for comparison purposes), second- and third-order fixed-order HMMs (i.e. models which do not have duration- or context modelling capabilities), second- and third-order context-emphasised HMMs, second- and third-order duration-emphasised HMMs as well as mixed-order context-duration-emphasised HMMs.

The higher-order HMM configurations were obtained and trained through application of the ORED/FIT algorithms [12]. The language models had separate transition probability networks, but shared the same set of state output probability densities (emitting densities). The classified language was the one which produced the highest log likelihood on the test utterance.

On the NIST 1995 language identification evaluation, Du Preez and Weber obtained a best RER of 19,4% on the 5s duration utterances and 2,6% on the 45s duration utterances performing pairwise language recognition of English and Hindi. The model which performed the best on the 5s utterances was a third-order duration modelling HMM, while the fixed order third-order HMM performed the best on the 45s utterances.

The RER obtained by first-order models on the 5s test utterances was 30,8% and on the 45s test utterances the first-order models obtained a RER of 17,9%. These results indicated that use of the higher order HMMs resulted in a significant reduction in RER for LID.

Accent identification

Chen, Huang, Chang and Wang utilised GMMs for identification of Mandarin Chinese accents [7]. The authors utilised MFCC features (and their derivatives) in conjunction with GMMs for the accent modelling. The influence of GMM component count as well as speaker gender on accent identification was investigated. The influence of the test segment duration on accent identification performance was also investigated.

The experimental results showed that the GMM mixture count had a significant effect on the accent identification performance of their system. An average RER of 17,6% was attained for the identification of four Mandarin Chinese accents, using average test segment durations of 16s. The Mandarin Chinese corpus contained about 16 hours of speech sampled at 16kHz for three of the accents, with one accent containing speech sampled at 22kHz.

Dialect identification

None of the published research reviewed for this literature synopsis performed dialect identification using a method based on spectral features only.

2.2.2 LC recognition utilising units of speech (phoneme, word etc.)

Language identification

As mentioned in subsection 2.1.2, the speech feature of choice for LID appears to be the phoneme. Accordingly, the majority of LID approaches utilised a phoneme recogniser front-end to tokenise the input speech as a stream of phonemes [41, 35]. Typically, the same phoneme recogniser was used for all the languages being recognised.

Often, the phoneme recogniser was not even trained on a language common to any of those being recognised. If only the frequency of phoneme occurrence is of importance, uni-gram language models were employed to model the languages. If phoneme context was also deemed to be of importance, N-gram language models were used to represent each of the languages under consideration [35].

The phoneme recogniser front-end combined with phonotactic language modelling forms the PRLM (Phone Recognition followed by Language Modelling) LID approach presented by Zissman and Singer [41]. Zissman et al proposed a number of improvements to the basic PRLM method. The first of these was PRLM-P, in which the phoneme recogniser front-end consisted of a parallel bank of phoneme recognisers.

For each language being recognised, the output from each of the front-end phoneme recognisers would be applied to the language's N-gram model and a log-likelihood score computed. The total language likelihood would simply be the summation of the log-likelihoods obtained for each of the different phoneme recognisers.

The second improvement to PRLM, PPR (Parallel Phoneme Recognition), appears to be a logical refinement of the PRLM/PRLM-P approach. PPR requires separate phoneme recognisers for each of the languages being recognised, with these phoneme recognisers only providing input for the N-gram model of the language on which they were trained.

The phoneme recognisers used for PPR differed from those used in PRLM in that right-context diphones were also utilised. Zissman et al compared GMM models, PRLM, PRLM-P and PPR by performing pair-wise language identification on selected languages

from the OGI-TS corpus. On average, the PPR method obtained the best RER, with the GMM models faring the worst.

Marcheret and Savic implemented a PPR-like configuration, but utilised language models based on random walk theory instead of the traditional N-gram language models [28]. As with the standard PPR configuration, each of the languages to be recognised had its own phonetic segmenter. The phonetic segmenters consisted of HSMMs (Hidden Semi-Markov Models), which are ergodic HMMs with discrete probability duration modelling and continuous observation probability modelling.

The random walk classifiers for a given language would produce Gaussian score distributions with zero mean for phonetic segments from the same language, but would produce non-Gaussian score distributions with non-zero mean for the other languages. For each series of phonetic segments input to the system, an Euclidean metric would be applied to the score distributions to determine which distribution had a mean closest to zero. This would then be the classified language. On the March NIST 95 language identification evaluations this method obtained a best pairwise RER of 10% on test segments longer than 30s and a RER of 12% on test segments of 10s length.

An improvement to the PRLM-P method which incorporated speaker gender information, was proposed by Zissman [42]. This basically entailed the training of separate gender-dependent as well as gender-independent phoneme recognisers for each of the languages used in the phoneme recogniser front-end. This resulted in three separate phoneme recognisers for each of the front-end languages, i.e. a phoneme recogniser trained on speech from male speakers only, a phoneme recogniser trained on speech from female speakers only and a phoneme recogniser trained on speech from both male- and female speakers. For each of the languages recognised, separate N-gram language models were trained for each of the gender-class phoneme recognisers. The acoustic likelihood scores produced by the male- and female phoneme recognisers were utilised to determine the probability of the test segment representing male speech.

Using this probability, weights were derived for each of the gender-specific channels. The language with the maximum likelihood summed and weighted gender-specific channel log-likelihood was then the hypothesised language. Additional performance gains were obtained by Zissman when phoneme duration modelling was added to the N-gram language models. Zissman used eleven-way forced-choice classification experiments on the OGI-TS corpus to show the improvement in PRLM-P performance with the addition of speaker gender information as well as phoneme duration statistics to the language model-

ling stage. On the 45s test segments the baseline PRLM-P system obtained an average RER of 20,3%. Incorporating speaker gender and phoneme duration information in the language model decreased the RER to 11,2% on the same test.

Further enhancements to the basic PRLM-P language identification algorithm were presented by Zissman [43]. The first improvement to the basic PRLM-P algorithm involved the addition of a Gaussian classifier back-end to the N-gram language modelling stage. The original PRLM-P algorithm used the linear, weighted combination of the log-likelihoods produced by the language models corresponding to each of the phoneme recogniser front-ends to determine the most likely language. The Gaussian classifier back-end involved the training of Gaussian probability densities for each of the languages being recognised, using the log-likelihood scores from the language model for each phoneme recogniser front-end as feature vector. This training was done during development testing.

Another improvement to the basic algorithm involved the use of phoneme frequency vectors (or VPF) to estimate Gaussian probability densities for each language which modelled the phoneme sequence probabilities for each language. The VPF vectors were computed during development testing by obtaining the phoneme count for each of the phoneme types used by the system, for a given development utterance, and then dividing the phoneme counts by the total number of phonemes in the utterance. Zissman deemed the multinomial phoneme sequence probability model used previously to be inadequate. On the NIST 1996 LID evaluation, the improved PRLM-P system obtained a RER of 25,7%, 46,6% and 65,2% on the 30s, 10s and 3s twelve-alternative, forced-choice test segments.

The method of Yan and Barnard followed the PRLM approach to perform nine-way language identification on the LDC Conversational Telephone Speech Database [39]. Separate phoneme recognisers were used for each of the nine languages, with the actual language modelling consisting of N-gram language models with duration modelling. The final classifier consisted of a feed-forward neural network with one hidden layer. This was the same system used in previous work of the authors [6], but applied to conversational speech instead of constrained speech as used before. Since this system was not trained on conversational speech but rather constrained speech from the OGI-TS corpus, a dramatic increase in the error rate was observed for the system.

The authors proposed two methods to overcome this problem. The first method utilised an energy criterion to discard long sections of silence as well as cepstral mean subtraction to overcome the channel mismatches between the training and testing environments. The

second method imposed constraints on the valid phoneme combinations output by the phoneme recognisers, based on the difference in phoneme recogniser output as a result of using conversational speech versus the constrained speech on which the system was trained. The baseline system, trained on constrained speech from the OGI-TS corpus, obtained a RER of 40,3% on the unconstrained speech of the Callfriend corpus for a nine-way LID experiment. A system which incorporated the improvements suggested by Yan and Barnard, reduced the RER for the same experiment to 23,6%.

Metze, Kemp, Schaaf, Schultz and Soltan used a word lattice confidence measure for recognition of English, German and Japanese speech on the VERBMOBIL multilingual speech-to-speech translation system [29]. Instead of using the classification scores output by the system directly, they utilised a lattice representing the words occurring in the test utterances, with one word at each lattice node. The lattice was interpreted as an HMM, with the word acoustic scores representing the HMM output densities and the lattice structure the HMM link structure. The HMM transition probabilities were obtained from tri-gram language model probabilities. It would appear as if a separate lattice was utilised for each language.

For a test utterance word sequence, the word lattice HMM was utilised to obtain the word sequence probabilities (or the ‘confidence measure’ as the authors referred to it), for each language. This ‘confidence measure’ was used to determine the most likely language. Previous work by the authors indicated that word-based LID systems outperformed phoneme based systems. It was shown by the authors that channel effects severely affect the recognition performance of LID systems which utilise a simple score-based approach. The utterances used for testing had an average duration of 7,9s. The score based approach yielded an overall LID RER of 7,2% for three two-way LID experiments. Using the confidence based LID approach, the overall LID RER for the same three two-way LID experiments was reduced to 6,4%.

A relatively successful LID approach has been the large vocabulary continuous speech (LVCSR) recognition LID system. This type of LID system combines phonetic, phonotactic, lexical and syntactic-semantic information for the overall system. Schultz, Rogina and Waibel reported on the use of LVCSR-based LID systems [38]. Their LID system consisted of the basic PPR LID system, with the difference that the scores obtained from each of the independent language models were fed into a multi-layer perceptron (MLP) neural network classifier back-end. The development and testing of the system was performed with the Spontaneous Scheduling Task (SST) corpus. Five different systems were investigated to determine the relative contributions of each of the knowledge sources used

in LVCSR LID.

The first system utilised continuous density HMM context-independent phoneme recognisers for each of the language model front-ends. The second system included phonotactic information in the form of bi-gram counts. The third system was a word-based recogniser incorporating the rules for phoneme concatenation to form words. The fourth system incorporated word bi-grams (or ‘word knowledge’ as the authors called it). The final system did not incorporate the language modelling directly, but applied a scoring routine to the word bi-gram output of the fourth system to hypothesise the most likely language. Three two-way LID experiments were conducted for each of the five LID configurations in turn. The best average two-way LID RER of 6,8% was achieved by the final system (scoring word bi-gram output from system four). The best four-way LID RER achieved by the authors was 16%.

The LVCSR LID system of Hieronymus and Kadambe used a variation of the basic PPR method [16]. The authors used continuous density variable duration second-order ergodic HMMs with one phoneme per state for the phoneme recogniser front-end of each language. This enabled them to recognise triphones (or ‘context-conditioned phonemes’ according to the authors) instead of the context-independent phonemes used by other PPR implementations. According to the authors, this is of great importance if the language syllable structure is of the CVC (consonant vowel consonant) form.

The authors used MFCC features with their first- and second-order derivatives. The phoneme recogniser front-ends were trained on phonetically hand-labelled speech. The language models consisted of N-gram language models with additional word-level lexical constraints. The authors developed a score normalisation procedure by which the utterance log-likelihood was normalised by the log-likelihood of the best unconstrained phoneme sequence. This normalisation technique in conjunction with cepstral mean normalisation resulted in a best five-way language identification RER on the OGI-TS corpus of 2% and 7% for 50s and 10s test utterances respectively.

Accent identification

PPR (in slightly modified form) was used by Kumpf and King for the identification of accented Australian English [23]. A phoneme recogniser trained on the so-called ‘native’ Australian English accent was used to segment the input speech into phoneme labels. Phoneme recognisers were then trained for the remaining two accents using the generated phoneme labels for each accent. The N-gram accent models were derived from the phoneme labels of the accent in question, according to the basic PPR approach. The

authors also conducted experiments to compare the accent identification accuracy when hand-segmented labels were used instead of the automatically segmented labels. Using 3150 test utterances of 4s average duration each on their PPR system, Kumpf and King obtained a three-way accent classification RER of 23,4%.

Two different accent identification methods were proposed by Lincoln, Cox and Ringland [25]. The first of these was phonotactic accent identification which utilised bi-gram models (diphones) in conjunction with confidence thresholds for the classification of British- and American English. These confidence thresholds related overall diphone frequency of occurrence to the recogniser input diphone history. Positive accent identification would only be made once the threshold corresponding to a specific accent was crossed. Input speech was tokenised into a phoneme stream using HMM monophone recognisers.

The second method proposed by Lincoln et al was the Mixture Component Usage (MCU) method. This method pooled speech from both accents to train monophone models and a silence model. The monophone models were SCHMM (Semi-Continuous HMM) models. The SCHMMs (by definition of the SCHMM) utilised the same set of HMM state output densities. During training of the monophone models, for each speaker, the most likely HMM state and mixture component for each input frame was noted.

Following training of the monophone models, each speaker had associated with him or her a vector indicating the most likely mixture component for all monophone HMM states. These vectors of mixture component usage formed a new pattern space from which a clustering of users according to accent was obtained using distance measures. During testing, the MCU vectors for the input speech is computed and using the appropriate distance measures, classified as belonging to one of the accents represented by one of the earlier MCU vector clustering centroids. Unfortunately no concrete RER figures were available for the accent identification methods proposed by Lincoln et al, but from the graphical classification results summary it appears that the method provided acceptable performance on both the database used for system development as well as on the independent TIMIT database.

Fung and Kat utilised phoneme-classes instead of phonemes for accent identification [21]. The TIMIT and HKTIMIT corpora were utilised for these experiments. It is not entirely clear from the paper how the accent modelling was performed. It is mentioned that a sequence of three-state HMMs with single Gaussian state densities were used, but these were probably for the phoneme classes and not the accents being recognised. It is assumed that the best RER of 14.52% presented in the paper was for the pairwise accent identifi-

cation of American English (represented by TIMIT) and Cantonese English (represented by HKTIMIT).

Dialect identification

Rekart, Zissman, Gleason and Losiewicz performed pair-wise dialect identification of Cuban- and Peruvian Spanish using the basic PRLM configuration [35]. A minimum of 460 minutes of speech was available for training each of the dialect models, and similar duration speech was available during testing. The speech for this experiment was sampled at 48kHz using 16-bit quantisation. Rekart et al obtained a RER of 16% on the pairwise identification of the two Spanish dialects, using training and testing speech segments of three minute duration.

The use of so-called ‘shibboleth’ words to perform automatic clustering of speakers into distinct groups sharing the same speech pronunciation characteristics was proposed by Huggins and Patel [19]. Here the emphasis was not on dialect recognition per se, but to improve the robustness of a command- and control system to the influence of dialectal variability. The relative frequency of occurrence for specific phonetic realisations of key-words (the ‘shibboleth’ words) were used to setup distance metrics which determined the automatic clustering of speakers into distinct dialect groups. A phoneme recogniser front-end supplied the phoneme stream from which different realisations of the ‘shibboleth’ words were identified.

2.3 Conclusions drawn from the literature synopsis

The literature synopsis indicated that the general LC recognition trend is to utilise one of the following speech recognition configurations:

- The first configuration utilises one or more phoneme recogniser front-ends, followed by language modelling based on the phoneme stream output by the recogniser front-end. Often the LC modelling is based on N-gram language models.
- The second configuration models the LCs directly in terms of the speech spectral features. I.e. no speech transcriptions are utilised for this type of LC recognition configuration. This type of configuration typically models the LCs with GMMs or ergodic HMMs.

Chapter summary

This chapter presented a brief review of previous research conducted in LC recognition. The purpose of this review was to determine the general trend in LC recognition research. This trend gives us a good indication of how to tackle SAE SV classification. The following chapter describes the AST SAE corpus.

The AST South African English (SAE) corpus

Introduction

This chapter introduces the AST South African English (SAE) corpus, which was developed and utilised in this thesis. The following topics are discussed in this chapter:

- Description of the SAE SVs (section 3.1).
- Collection of the SAE corpus speech (section 3.2).
- SAE corpus speech statistics (section 3.3).

3.1 The SAE SVs

Each SAE SV is identified by two letters, with the first letter denoting the first language of the speakers and the second letter denoting the base language of the SV (SAE). The following five SAE SVs were identified at the onset of the AST project:

- Afrikaans English (AE)
AE denotes English spoken by people who have Afrikaans as their first language. It is understood that the Afrikaans spoken by these people is so-called 'standard' Afrikaans.
- Black English (BE)
English speech from speakers of the Nguni and Sotho language families was placed in the BE SV. For the AST project, the Xhosa and Zulu languages were chosen to represent the Nguni family, whereas SoSotho was the language chosen to represent the Sotho family.

Chapter 3

The AST South African English (SAE) corpus

Introduction

This chapter introduces the AST South African English (SAE) speech corpus which was utilised in this thesis. The following topics are discussed in this chapter:

- Description of the SAE SVs (section 3.1).
- Collection of the SAE corpus speech (section 3.2).
- SAE corpus speech statistics (section 3.3).

3.1 The SAE SVs

Each SAE SV is identified by two letters, with the first letter denoting the first language of the speakers and the second letter denoting the base language of the SV (SAE). The following five SAE SVs were identified at the onset of the AST project:

- **Afrikaans English (AE)**

AE denotes English spoken by people who have Afrikaans as their first language. It is understood that the Afrikaans spoken by these people is so-called 'standard' Afrikaans.

- **Black English (BE)**

English speech from speakers of the Nguni and Sotho language families was placed in the BE SV. For the AST project, the Xhosa and Zulu languages were chosen to represent the Nguni family, whereas SeSotho was the language chosen to represent the Sotho family.

- **Coloured English (CE)**

The **CE** SV represents English spoken by people from the Cape Flats area of South Africa's Western Cape Province. The first language of these speakers could be English or Afrikaans, but the Afrikaans and English spoken by this community differs sufficiently from standard Afrikaans and standard South African English to warrant separate Afrikaans and English SVs for these speakers.

- **English English (EE)**

English spoken by people who have English as their first language and which cannot be categorised as one of the other SAE variants, was placed in the **EE** SV. The 'English English' (**EE**) SV group represents 'standard' South African English. Historical factors dictate that standard South African English is modelled after British English.

- **Indian English (IE)**

English spoken by people of Indian (mainland India) descent.

3.2 SAE corpus speech collection

The SAE corpus consists of telephone calls from land lines as well as cell phones, collected over ISDN (Integrated Services Digital Network) lines. A Dialogic D/300-SC ISDN interface board was used for the call recording. The caller utterances were sampled at 8kHz and recorded to single-channel 8-bit alaw format speech files, with separate speech files for each utterance [1].

The callers were provided with numbered prompt sheets. The call sheet prompts were designed to elicit¹ both constrained- and spontaneous speech. The speech files were processed to determine the quality (usability) of the recorded utterances. This quality assessment stage was responsible for removing unusable utterance speech files.

The quality or usability of the utterances was influenced by the presence of non-speech events in the recorded utterances. These non-speech events ranged from caller-generated noises (such as lip-smacking by the caller in between speaking) to background noise generated by machine equipment etc. Following the quality assessment, the utterances were transcribed orthographically.

¹The Pocket Oxford Dictionary [3] entry for 'elicit' reads as follows:

"...draw out (facts, a response, etc.), esp. with difficulty."

The orthographic transcriptions contained only the onset and termination times of the utterance phone stream. The time of occurrence for non-speech events was also indicated in the speech transcriptions. As mentioned in the previous chapter, the speech transcription process was not finished when the thesis implementation stage commenced and therefore speech transcriptions were not used for this thesis. Also; many of the speech files were not yet processed by the quality assessment personnel when the thesis implementation stage commenced.

The telephone type (channel) as well as caller gender were therefore unknown for many of the AST SAE speech files. The consequence of this was that separate speech recognition models for the two telephone channel types and the two genders could not be developed. Using separate speech recognition models for the telephone channel type and the caller gender could improve the overall robustness of the SAE classification system.

This is because the caller gender determines the spectral characteristics of the speech to a significant degree and the different telephone channel types have different transfer functions which shape the spectral characteristics of the recorded speech in different ways. It is anticipated that mismatches between the speaker gender- and telephone channel characteristics of the SVs will impact the classification process negatively.

Gender- and telephone channel type specific modelling of the SVs might utilise the following models for each SV:

- A model for *female* speakers using *cell phones*.
- A model for *female* speakers using *landlines*.
- A model for *male* speakers using *cell phones*.
- A model for *male* speakers using *landlines*.
- A model for all the SV speech (both genders and telephone channel types).

Classification will then proceed as normal, where each test segment is simply matched to all the SV models, and the SV's model which produces the highest log-likelihood score is assigned the test segment; irrespective of which gender/channel model of the SV produced the best score.

The use of gender/channel-specific models would also expand the classification possibilities for the SV classification system. For criminal forensics speech recognition applications it would be useful to determine the probable gender of the caller and the type of telephone

utilised by the caller.

On the other hand, by not splitting the AST SAE corpus according to caller gender and telephone channel type, one has more speech with which to train the SV speech models. The duration of speech available for training of the SV speech models is an important consideration for the training of higher order HMMs. An increase in HMM order results in an increase in the duration of speech required for satisfactory parameter estimation of the HMMs.

Garbage or unusable utterance files were discarded from the speech file collection as these were discovered. These unusable utterance files caused the speech recognition process to fail, and were at first discovered by a process of elimination. The speech signal for many of the unusable utterance recordings was in fact a DC signal with zero amplitude. These 'empty' utterance files were probably the result of the speech recording software recording when there was no input signal to the ISDN interface board. Needless to say, empty utterance files are of no use to the speech recognition development process and will in fact have a detrimental effect on the estimation of speech model parameters etc.

The speech files need not be completely empty to create problems for the speech recognition system development process. Zero amplitude sections of the input speech which have duration equal to or longer than the frame length used for the computation of speech features, cause numerical instabilities in the computation of speech features. A simple speech preprocessor was implemented to remove these problematic zero amplitude sections of the input speech. This preprocessor, the Constant Amplitude Removal Preprocessor (CARP), is presented in subsection 4.1.3. The CARP enabled the speech recognition system to automatically discard empty utterance files.

Many of the SAE utterance files have long sections containing only background noise, with no speech input from the caller. This is caused by a number of factors, such as inadequate speech endpoint detection by the speech recording system as well as the presence of automatic gain control (AGC) units in the speech recording system. These problems are addressed in subsection 4.1.4. To combat these problems and hence improve system robustness to non-speech events, the Power Floor Preprocessor (PFP) was implemented. The PFP basically segments an input speech signal into speech and non-speech portions based on a morphologically filtered version of the input speech signal's frame power. The design and detail of the PFP algorithm is presented in subsection 4.1.4.

3.3 SAE corpus speech statistics

Figure 3.1 shows the amount of speech available for each of the SAE SVs. This figure shows the total speech duration, as well as the separate durations for the training and testing utterances of each SV. These figures may be different for the final version of the SAE corpus, but they represent the set of SAE corpus utterances used in this thesis. A detailed summary of the SAE corpus speech statistics is found in Table A.1. As mentioned in the previous section, the telephone type as well as the caller gender was unknown for many of the recorded utterances. This is reflected in Figure 3.1 and Table A.1.

The bars labelled ‘Unknown’ in Figure 3.1 indicate the total utterance duration of each SV for which the caller gender and telephone type is unknown. The telephone type as well as the caller gender was unknown for 78% of the utterances utilised in this thesis. This percentage is obtained by measuring the duration of the ‘Unknown’ utterances vs. those for which the caller gender and telephone type is known.

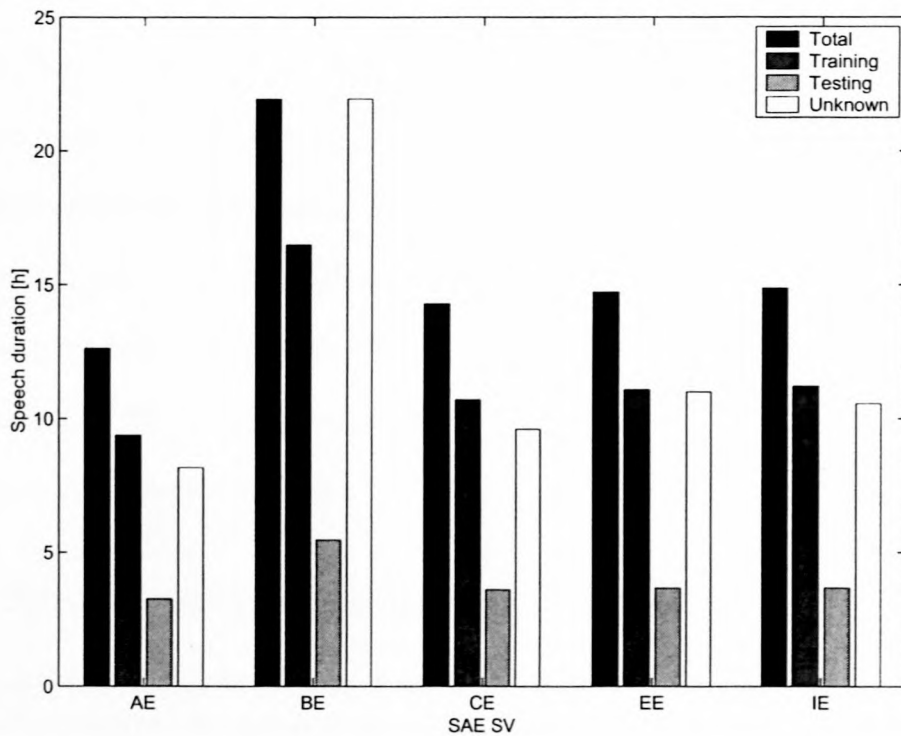


Figure 3.1: *SAE corpus speech durations. Shown for each SV is the total duration speech available for the SV (‘Total’), the total duration of the training utterances (‘Training’) and the total duration of the utterances used for testing (‘Testing’). Also shown for each SV is the total duration of speech for which the caller gender and telephone type was unknown (‘Unknown’).*

Chapter summary

This chapter introduced the speech database which was the focus of the thesis, namely the AST SAE corpus. The SVs which comprise this corpus were briefly described. The speech collection process and some of the corpus speech statistics were presented in order to illustrate the corpus characteristics. The following chapter will present the theoretical concepts utilised in this thesis.

Theory

Introduction

This chapter presents the basic signal processing and pattern recognition concepts used in the thesis. It is not meant to be an exhaustive treatment of these concepts. The following topics will be discussed:

- Speech preprocessing (section 4.1).
- Speech features (section 4.2).
- Feature normalisation (section 4.3).
- Vector Quantisation (VQ) (section 4.4).
- Speech modelling (section 4.5).
- Hierarchical Vector Quantisation (HVQ) (section 4.6).
- Speech classification (section 4.7).

4.1 Speech preprocessing

The aim of the speech preprocessing stage is to compensate for degradation of the input speech and to enhance those speech characteristics parametrised by the speech features. Any speech degradation or artifact not dealt with during the speech preprocessing stage will propagate through the speech recognition system and degrade the system performance. The speech preprocessors used in this thesis will be discussed next.

Chapter 4

Theory

Introduction

This chapter presents the basic signal processing and pattern recognition theory utilised in the thesis. It is not meant to be an exhaustive treatment of these concepts. The following topics will be discussed:

- Speech preprocessing (section 4.1).
- Speech features (section 4.2).
- Feature normalisation (section 4.3).
- Vector Quantisation (VQ) (section 4.4).
- Speech modelling (section 4.5).
- Hierarchical Vector Quantisation (HVQ) (section 4.6).
- Speech classification (section 4.7).

4.1 Speech preprocessing

The aim of the speech preprocessing stage is to compensate for degradation of the input speech and to enhance those speech characteristics parameterised by the speech features. Any speech degradation or artifact not dealt with during the speech preprocessing stage will propagate through the speech recognition system and degrade the system performance. The speech preprocessors used in this thesis will be discussed next.

4.1.1 Preemphasis

Preemphasis involves all-pole filtering of the input speech, which results in emphasis of the high-frequency components. It also assists in removing the influence of the larynx and lips on the speech production process and hence provides better isolation of the vocal tract response. The all-pole filter polynomial is given by [9]:

$$P(z) = 1 - \mu_p z^{-1}, \quad (4.1)$$

with the preemphasis coefficient μ_p in the range $0.9 \leq \mu_p \leq 1.0$.

4.1.2 Power normalisation

Power normalisation caters for varying recording volumes, and involves the following:

- The input speech is divided into blocks.
- The power is computed for each of the blocked sections.
- The blocked speech sections are divided by the square root of the 75th percentile of the power values computed for all the blocks.

The 75th percentile was chosen as a result of practical experience with power normalisation. It was found that use of the median of the blocked power levels caused the preprocessor to suppress significant silence sections of the input speech. If the maximum blocked power level was used, it resulted in a bias toward background noise with high amplitude, such as the noise caused by a slamming door.

4.1.3 Constant Amplitude Removal Preprocessor (CARP)

Constant amplitude sections are parts of the input utterance waveform which have zero variance. If these constant amplitude sections have zero amplitude and are of duration equal to or greater than the feature extraction analysis frame length, numerical instabilities may occur during the feature calculation process. This is especially true if the feature extraction process involves the computation of logarithms.

To overcome the problem posed by long-duration zero amplitude speech sections, the CARP was devised. This preprocessor considers fixed-duration sections of the input speech waveform and determines whether any signal variation occurs over the section under consideration. If no variation is found, the section is removed from the input speech. This necessitates re-alignment of time-aligned speech transcriptions (if these are to be used), following application of the CARP.

The parameters for this preprocessor are:

- win_{dur} . The analysis window length (in seconds) of the CARP. The input speech waveform is divided into sections of length win_{dur} . These sections are tested separately for amplitude variation.
- ret_{dur} . The duration (in seconds) of discarded speech sections to retain. Short-duration silence sections often need to be retained in order to maintain inter-phoneme context.

The CARP was utilised for the automatic removal of so-called ‘empty’ utterance files. These utterance files contained no amplitude variation and hence no speech.

4.1.4 Power Floor Preprocessor (PFP)

Listening tests performed on randomly selected utterance files from the SAE corpus revealed that some of the speech files were severely corrupted by the presence of non-speech artifacts. The presence of non-speech artifacts in speech used to train SV models could lead to inaccurate modelling of the SV speech.

The speech endpoint detection (SED) performed in standard speech recording equipment simply detects the onset and termination of speech input, or input resembling speech. Non-speech artifacts could be recorded as speech if the SED is flawed or if the non-speech artifacts overlap with speech sections. The PFP was designed to overcome the former problem, whereas the latter problem was considered to be beyond the scope of this thesis.

The following were deemed to be possible sources of non-speech artifacts:

- The environment in which the call originated.
- The recording channel and equipment used to record the speech.
- Caller behaviour or speaker-specific noise.

The factors identified above did not necessarily occur independent of each other. Often the determination of a non-speech artifact’s cause was done in a subjective manner.

The above-mentioned factors correspond with those identified by Rabiner and Juang as factors which could hamper SED [20]. In their discussion of SED, Rabiner and Juang used a simple isolated digit recognition experiment to illustrate the increase in speech recognition error as a result of SED error. An important point made by the authors is

that the degree to which SED error will detract from the speech recognition system performance will probably depend on the speech recognition problem itself and the speech recognition system setup and characteristics.

According to Rabiner and Juang, the non-speech artifacts are often of the same energy content as speech events. It follows then that a SED approach which is simply energy-based will have limited success.

Influence of caller environment on the speech recording process

Some of the non-speech artifacts attributed to the caller environment were:

- Speech from external (non-participating) speakers. This type of speech sometimes happened to be of a different SV origin than that of target (speaker) SV.
- Animal noises (dogs barking in the background etc.).
- Machine equipment noise.
- Vehicle traffic noise.

The telephone channel and recording equipment as sources of non-speech artifacts

The AST SAE corpus included speech collected over land lines as well as cell phones, with both these telephone channels possibly generating distinct non-speech artifacts.

The non-speech artifacts generated by the telephone channel included the following:

- Call connect- and disconnect noise.
- Channel interference or call breakup (deterioration of speech beyond the point of intelligibility).

The speech artifacts introduced by recording equipment (telephony- and sound cards) included the following:

- Zero amplitude non-speech sections.
- Inability of recording equipment to reject or suppress non-speech sections of the input signal as a result of inadequate SED.
- Varying signal power levels caused by Automatic Gain Control (AGC) units in the recording equipment.

Automatic Gain Control (AGC) units are amplifiers which modify their gain such that the output signal of the unit has an amplitude greater than or equal to a predetermined threshold [17]. This might result in non-speech artifacts of significant power which could be mistakenly classified as speech by SED units.

Influence of caller behaviour on the recording process

It is not entirely sure to what extent speaker behaviour influenced the classification results obtained in this thesis. Sources of non-speech artifacts attributed to caller characteristics include the following:

- Speech impairments such as lisping and stuttering.
- Speaker idiosyncrasies which include the smacking of lips or noisy physical gestures used for emphasis.
- Speaker interaction with the handset. If the speaker handles the handset frequently it could lead to excessive noise contamination of the speech.
- Extremely long pauses (hesitation) by the caller between reading utterances.

The longest duration speech file in the SAE corpus had a duration of ten minutes. Only the first eight seconds of the file contained actual speech. The rest of the file consisted of background noise and noises made by the caller. This is a good example of inadequate SED combining with caller behaviour to produce an utterance file consisting mainly of non-speech artifacts. Needless to say, this file and others like it will definitely hamper speech recognition efforts unless they are dealt with.

Design and implementation of PFP

One would expect the power in speech signals as measured over successive frames, to vary between the power level corresponding to no input signal (so-called ‘ambient power floor’) and that corresponding to the maximum amplitude section of the input signal; assuming a sufficient signal to noise ratio (SNR) for the telephone channel and recording equipment. The resultant vector obtained from measuring the signal power over successive frames will subsequently be referred to as the ‘frame power’, or \mathbf{p}_{frm} of the signal.

The frames are obtained by dividing the input signal into overlapping sections of constant length (duration) and constant spacing (or ‘skip’). A frame length of 20ms and half-frame spacing (10ms) was used for this thesis. \mathbf{p}_{frm} is computed as follows:

$$\mathbf{p}_{frm}(f) = \log \left[\frac{\sum_{k=1}^N \mathbf{v}_f(k)^2}{N} \right], \quad (4.2)$$

where $\mathbf{p}_{frm}(f)$ is the resultant frame power for frame f , $\mathbf{v}_f(k)$ is the k 'th sample of \mathbf{v}_f , the speech samples corresponding to frame f ; and N is the number of samples in frame f .

The presence of AGC units in the recording equipment resulted in the amplification of non-speech artifacts or even silence sections to such an extent that these sections of the input signal had sufficient power levels to be confused with speech sections. This effect was manifested in varying ambient power floor levels of the recorded signal. Other non-speech artifacts, such as background noise generated by machine equipment, have power levels comparable to that of speech portions of the input signal and are obvious sources of confusion for any speech recognition system.

Another factor to consider for adequate SED is the limit imposed by the speech production system on the rate of change for a speech signal and its corresponding \mathbf{p}_{frm} . Typical vowel durations are around 40-400ms [9]. Sections of the input speech which are shorter than these typical vowel durations could be rejected, even if these speech sections have high enough power levels to be classified as speech by standard SED techniques. This constraint placed on the rate of change of the input signal should result in the rejection of amplitude spikes and other recording artifacts, which might have power levels comparable to speech, but which are of too short duration to be classified as speech.

The PFP design combines power level constraints and the temporal constraints mentioned above for the segmentation of an input signal into speech and non-speech sections, at the frame power level. Sections of the input signal which are classified as non-speech are replaced by constant amplitude sections. The constant amplitude sections may then be removed by application of the CARP (as was done in this thesis).

The PFP algorithm will be presented next using the example utterance shown in Figure 4.1. The utterance shown in Figure 4.1 has a total duration of around 70s. Only the sections between 0s-3,5s (Figure 4.1 (a)) and 30,5s-33,5s (Figure 4.1 (b)) contain speech. The section between 68s-70,5s (Figure 4.1 (d)) contains only recording artifacts. The utterance shown in Figure 4.1 contains background noise generated by machine equipment.

The power level of the machine equipment noise was high enough for the SED of the recording equipment to fail. In other words, the recording continued even when no speech was input to the system. The spectrogram for the PFP example utterance is shown in Figure 4.2. The spectrogram for the complete utterance is shown in Figure 4.2 (a) and the spectrogram portions corresponding to the speech sections of interest identified above, in Figure 4.2 (b), (c) and (d). The energy level of the signal at a given time/frequency

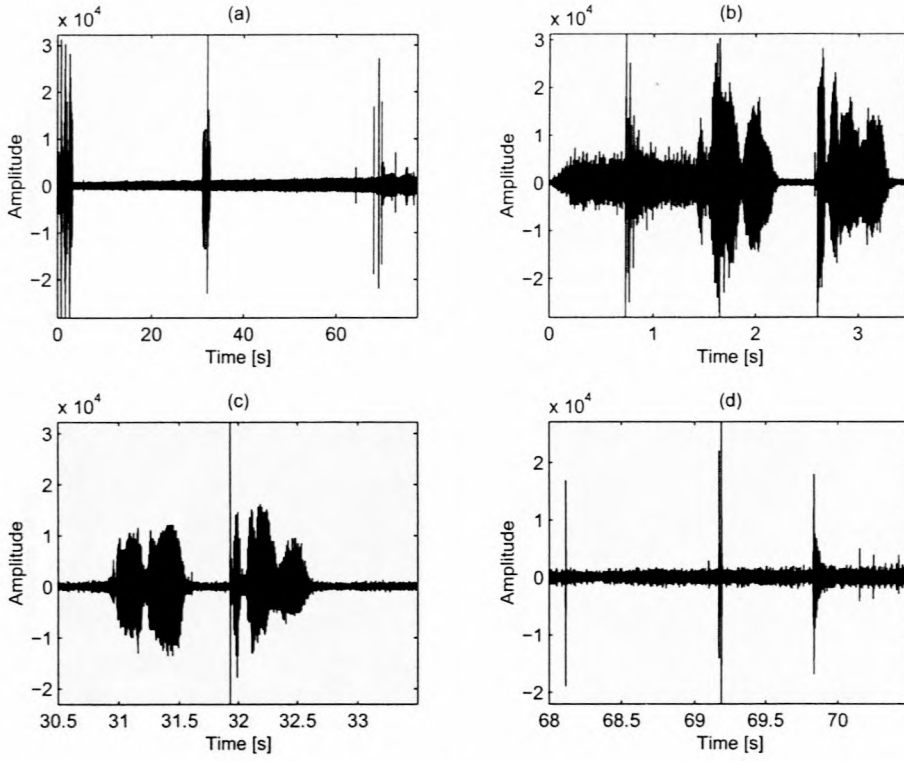


Figure 4.1: *Example utterance used for PFP algorithm description. The complete utterance is shown in (a); with (b), (c) and (d) zooming in on selected sections of the example utterance. Sections (b) and (c) of the utterance contain speech whereas section (d) contains only recording artifacts.*

coordinate is proportional to the spectrogram intensity level at that coordinate.

The presence of the machine equipment noise is more visible in the utterance spectrogram than in its speech signal. The machine equipment noise manifests itself in the spectrogram as invariant horizontal bands of relatively high energy; which occur primarily in the frequency range between 0-1kHz. The PFP algorithm consists of the following steps:

- *Computation of \mathbf{p}_{frm} using Equation 4.2.* The frame length frm_{dur} and overlap frm_{ovl} used for framing the input signal is specified beforehand by the user. The frame power computed for the example utterance, using a frame length of 20ms and a frame skip of 10ms, is shown in Figure 4.3. From the frame power for the complete signal, shown in Figure 4.3, one sees that the ambient power level for the signal increases gradually. This is attributed to the presence of AGC units in the recording equipment.
- *Filtering of \mathbf{p}_{frm} using a morphological ‘open-closing’ filter.* This morphological

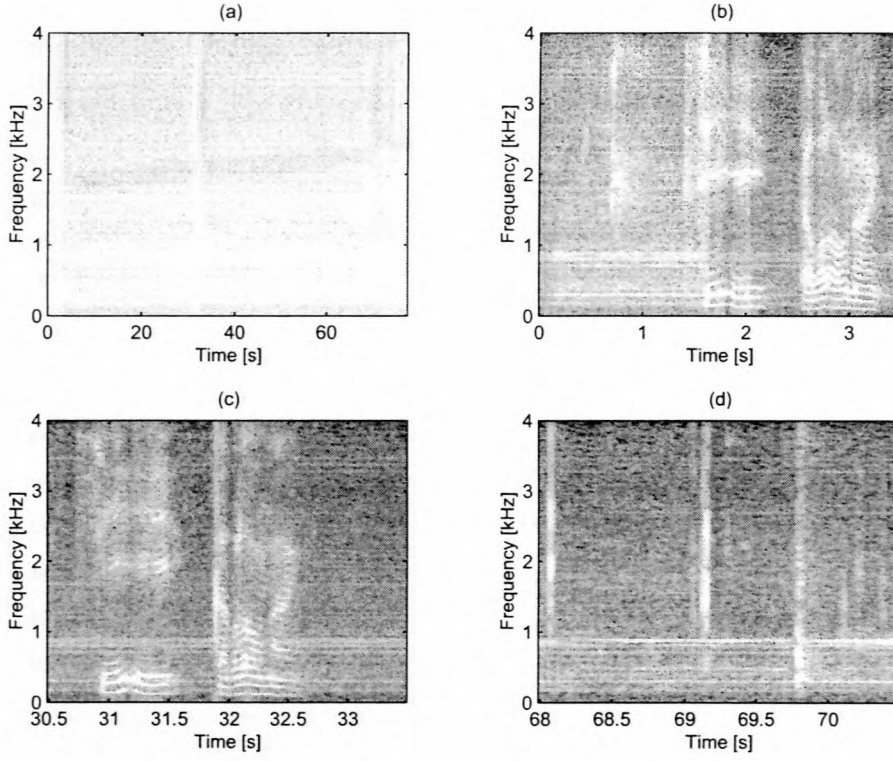


Figure 4.2: Spectrogram for the PFP example utterance (shown in Figure 4.1). Section (a) is the spectrogram for the complete utterance shown in Figure 4.1 (a). Sections (b), (c) and (d) are the spectrogram sections for the corresponding sections of Figure 4.1.

filter consists of a number of cascaded dilation and erosion filtering operations [14].

The morphological ‘opening’ filter removes peaks in \mathbf{p}_{frm} with duration less than a user-specified minimum peak duration $peak_{min}$, by flattening the peaks and not by discarding the peak samples.

The morphological ‘closing’ filter fills in the gaps (potential silence sections) between adjacent sections of \mathbf{p}_{frm} which correspond to speech sections of the input signal. Gaps in the frame power waveform of duration smaller than a user-specified minimum gap duration gap_{min} are filled in.

In other words; the morphological filtering stage of the PFP algorithm forms the envelope of possible speech sections based on the temporal characteristics of \mathbf{p}_{frm} . The morphological filtering operation introduces a lag in \mathbf{p}_{frm} , which is compensated for prior to proceeding with the next stage of the PFP algorithm.

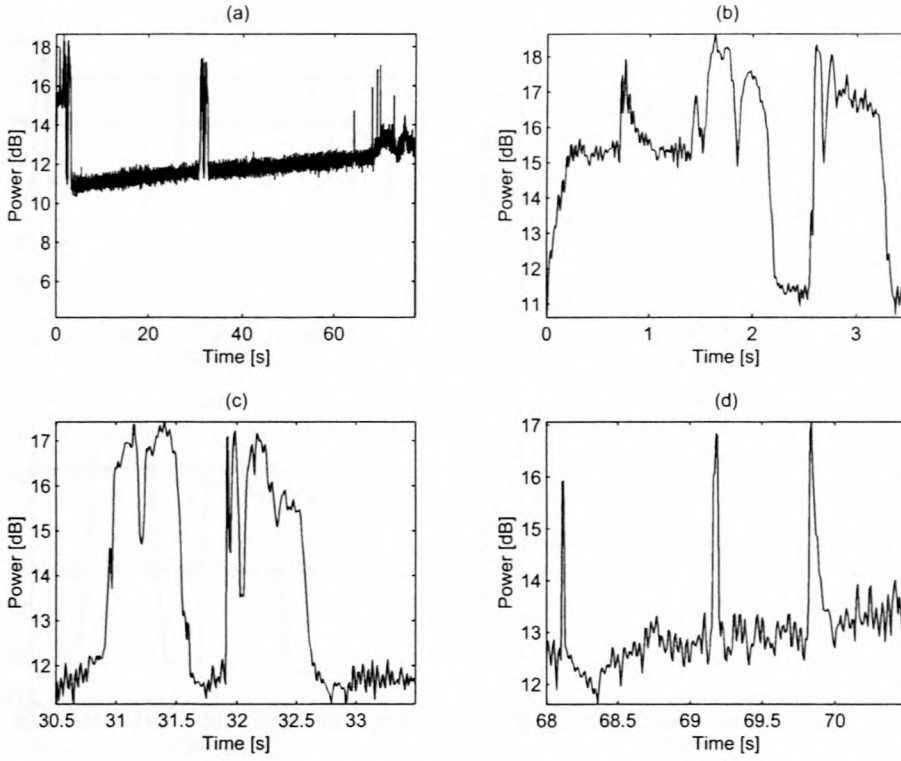


Figure 4.3: Frame power \mathbf{p}_{frm} for the PFP example utterance (shown in Figure 4.1). Section (a) of this figure is the frame power for the complete utterance, whereas sections (b), (c) and (d) represent the frame power for the corresponding sections of Figure 4.1.

To aid in the discussion of the PFP algorithm, the morphologically filtered version of \mathbf{p}_{frm} will be assigned the symbol \mathbf{p}_{morph} . The frame power for the example utterance, following the morphological filtering operation, is shown in Figure 4.4. The percentile levels and power level thresholds indicated in this figure form part of the percentile normalisation stage which will be discussed next.

- *Percentile normalisation of \mathbf{p}_{morph} .* This stage of the PFP algorithm refines the speech segmentation based on the power levels in \mathbf{p}_{morph} .

A percentile, symbolised by p_1 , of P_{morph} is computed to establish the ambient power floor for the input signal. The ‘1’ subscript indicates that it is the first such percentile computed as part of the percentile normalisation stage. For this thesis the second percentile was used for p_1 .

A different percentile would probably be required for a different speech corpus. The value of p_1 depends on the characteristics of the corpus speech to which PFP will be applied. Basically all of the PFP parameters are determined by inspection

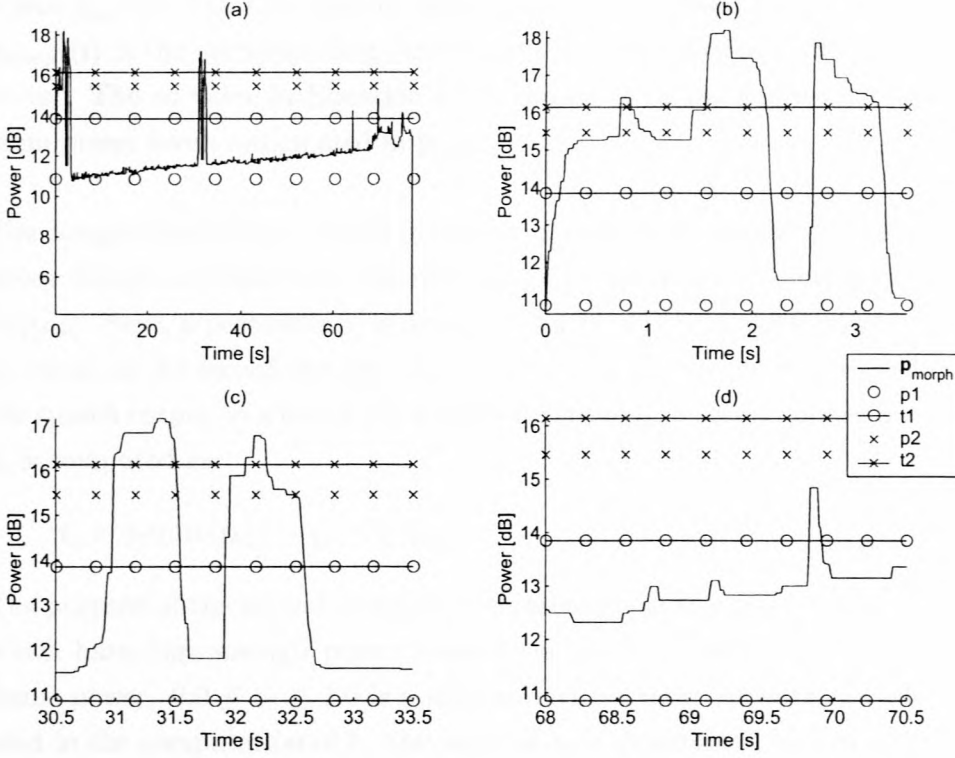


Figure 4.4: Morphologically-filtered frame power p_{morph} for the PFP example utterance. An ‘opening’ filter was applied to the frame power shown in Figure 4.3 to remove spikes. This was followed by application of a ‘closing’ filter to fill in potential silence sections. ‘p1’ and ‘p2’ are percentile levels which are utilised along with the threshold levels ‘t1’ and ‘t2’ to segment the utterance into speech and non-speech sections.

of the target corpus speech. Future research efforts could investigate automatic estimation of the PFP parameters.

From p_1 a power level threshold t_1 is computed:

$$t_1 = p_1 + o_1. \quad (4.3)$$

o_1 is a pre-defined offset which is determined by inspection of the frame power waveforms for a number of utterances from the corpus on which the PFP is to be applied. o_1 is chosen such that the threshold determined by t_1 is higher than the power level of utterance sections which are too low in power to be representative of speech. The threshold established by t_1 helps to reject soft background noises.

The first percentile normalisation is then performed as follows:

$$p_{p_1}(i) = \begin{cases} \infty & p_{morph}(i) \leq t_1 \\ p_{morph}(i) & p_{morph}(i) > t_1 \end{cases}, \quad (4.4)$$

where $\mathbf{p}_{p_1}(i)$ is the i 'th value of the percentile-normalised frame power vector and $\mathbf{p}_{morph}(i)$ is the corresponding element of the morphologically filtered frame power vector. The ∞ value in Equation 4.4 is chosen to be much greater than the maximum power levels anticipated in \mathbf{p}_{frm} .

The computation of \mathbf{p}_{p_1} should be able to remove non-speech sections of the frame power waveform which have relatively low power compared to actual speech sections of \mathbf{p}_{frm} . Next, a percentile p_2 is computed for \mathbf{p}_{p_1} . For the AST SAE corpus, p_2 was set equal to the second percentile of \mathbf{p}_{p_1} . As with p_1 , the value of p_2 will depend on the speech corpus to which PFP is applied. From p_2 , a second power level threshold t_2 is computed as:

$$t_2 = o_2 \max(\mathbf{p}_{p_1}) + (1 - o_2)p_2. \quad (4.5)$$

The purpose of the second power level threshold t_2 is to remove non-speech sections which have high enough power levels to be confused with speech sections of the frame power. $0.0 \leq o_2 \leq 1.0$ is a user-defined fraction. As with the offset value o_1 used in the computation of t_1 , the value of o_2 is determined by inspection.

This second threshold t_2 is especially useful in removing non-speech artifacts caused by AGC units. An example of using the PFP to remove AGC non-speech artifacts is presented right after the description of the PFP algorithm.

The new frame power \mathbf{p}_{p_2} is next computed as:

$$\mathbf{p}_{p_2}(i) = \begin{cases} p_2 & \mathbf{p}_{p_1}(i) \leq t_2 \quad \text{or} \quad \mathbf{p}_{p_1}(i) = \infty \\ \mathbf{p}_{p_1}(i) & t_2 < \mathbf{p}_{p_1}(i) < \infty \end{cases}. \quad (4.6)$$

From \mathbf{p}_{p_2} the final speech segmentation is obtained. The sections of \mathbf{p}_{p_2} which are not equal to p_2 represent frames which contain speech.

Listening tests established that the speech segmentation obtained in this fashion discards very soft speech sounds such as unvoiced fricatives. To overcome this problem, an additional parameter was introduced for the PFP.

This parameter, ret_{dur} ('retention duration') specifies what duration of \mathbf{p}_{p_2} adjacent to segmented speech sections should be retained to prevent the accidental removal of speech sounds which have low power levels. As with most of the parameters of PFP, some experimentation is required on the part of the user in determining a suitable value for ret_{dur} .

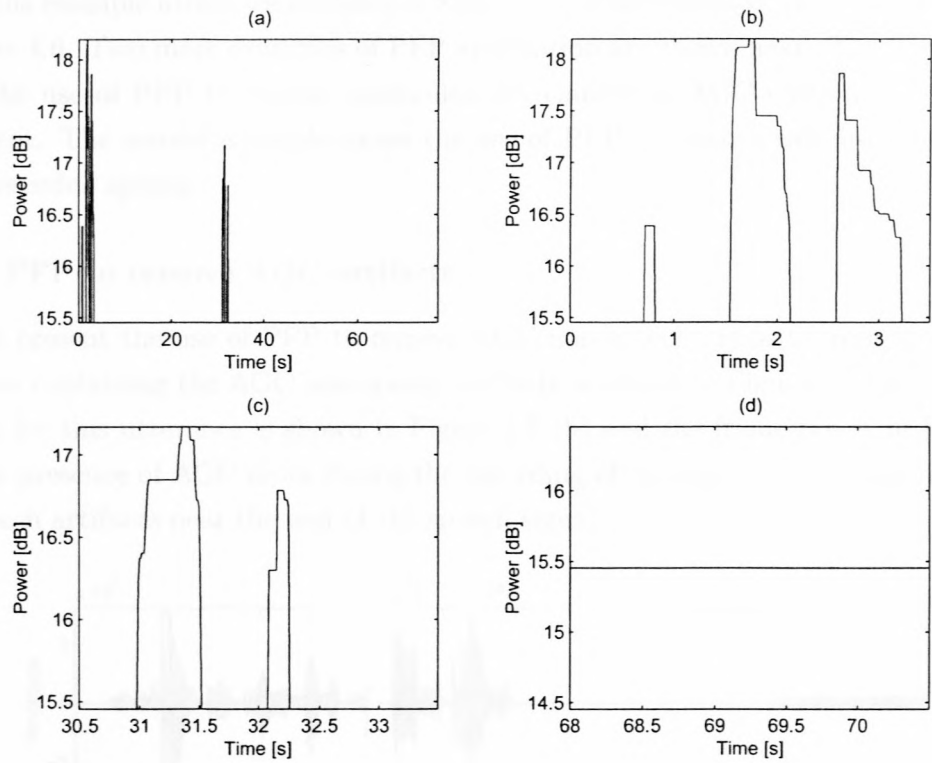


Figure 4.5: Percentile-normalised frame power p_{p_2} for PFP example utterance.

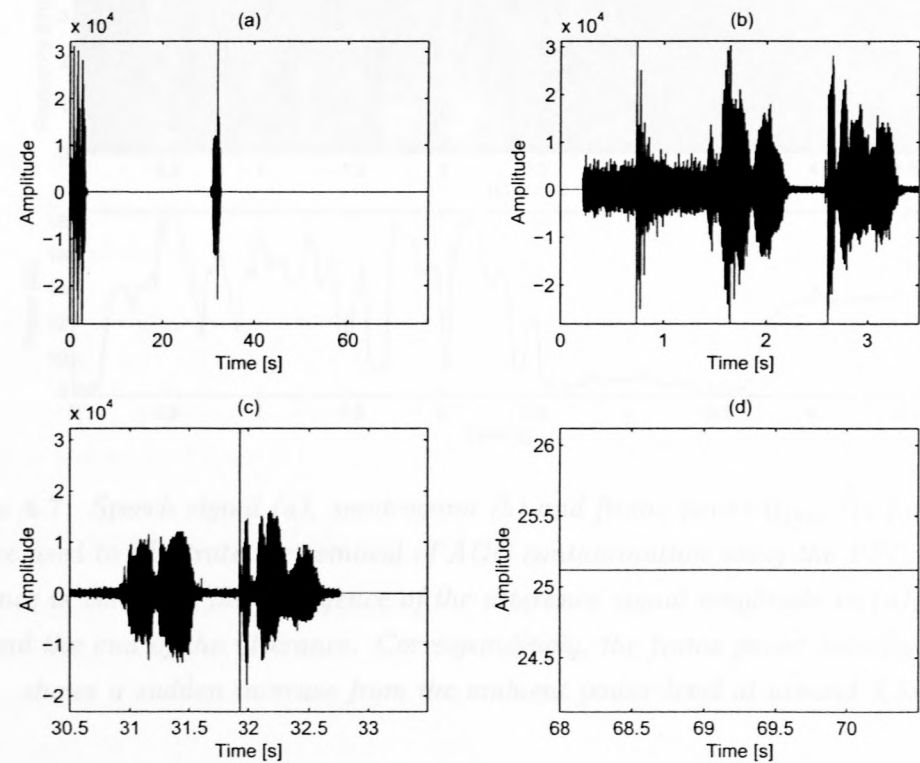


Figure 4.6: Example utterance following PFP application.

p_{p2} for the example utterance is shown in Figure 4.5. The processed speech signal is shown in Figure 4.6. Two more examples of PFP application are shown next. The first example shows the use of PFP to remove contamination caused by AGC units in the recording equipment. The second example shows the use of PFP to remove call disconnect spikes in the recorded speech.

Use of PFP to remove AGC artifacts

Here we present the use of PFP to remove AGC non-speech artifacts from speech. The utterance containing the AGC non-speech artifacts is shown in Figure 4.7 (a). The spectrogram for this utterance is shown in Figure 4.7 (b) and the frame power in Figure 4.7 (c). The presence of AGC units during the recording of the signal in question resulted in non-speech artifacts near the end of the speech signal.

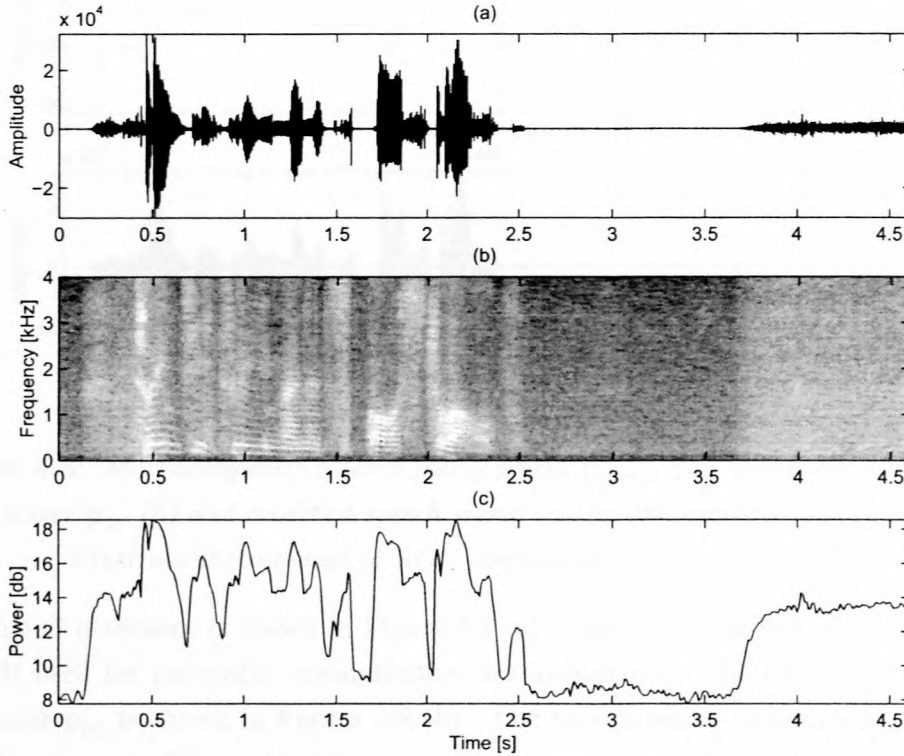


Figure 4.7: Speech signal (a), spectrogram (b) and frame power p_{frm} (c) for example utterance used to illustrate the removal of AGC contamination using the PFP. The AGC influence is visible in the divergence of the utterance signal amplitude in (a), between 3,5s and the end of the utterance. Correspondingly, the frame power waveform in (c) shows a sudden increase from the ambient power level at around 3,5s.

This contamination appears in the speech waveform as a sudden divergence in the speech signal amplitude, followed by random amplitude fluctuation. In the spectrogram of the speech signal, it manifests itself as noise. In the frame power waveform, the AGC contamination is indicated by a sudden jump in frame power level near the end of the frame power waveform. The morphologically-filtered frame power p_{morph} for the AGC-

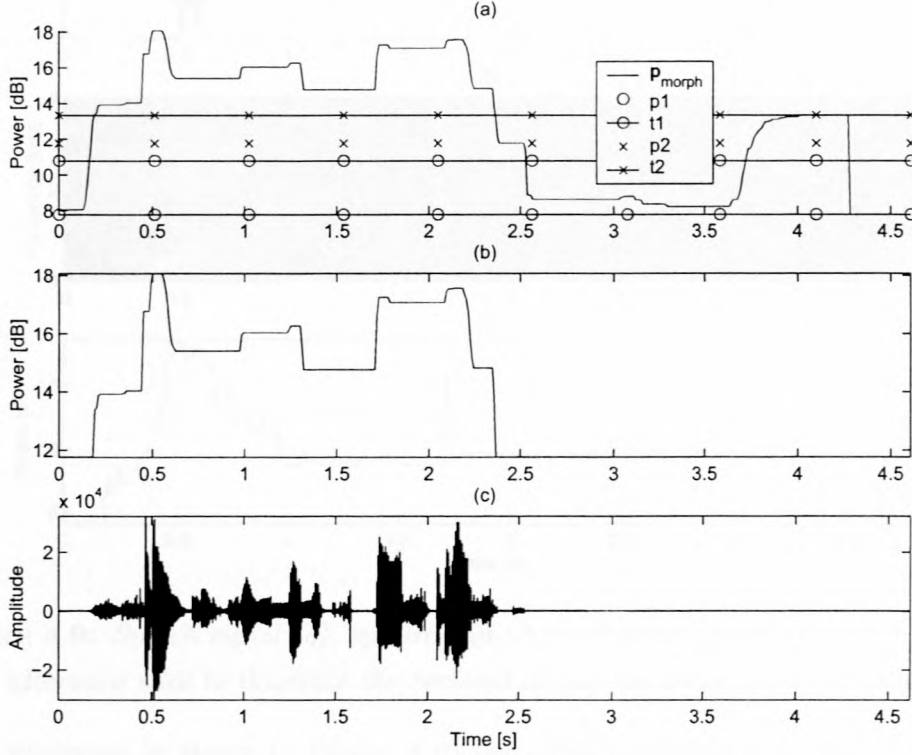


Figure 4.8: Morphologically filtered frame power p_{morph} (a), percentile-normalised frame power p_{p2} (b) and modified speech signal (c) for the example utterance used to illustrate the removal of AGC contamination using the PFP.

contaminated utterance is shown in Figure 4.8 (a). The percentile levels and power level thresholds used for percentile normalisation are also shown. The percentile-normalised frame power p_{p2} is shown in Figure 4.8 (b). The speech signal following application of the PFP is shown in Figure 4.8 (c).

Use of PFP to remove call disconnect spikes

Here we present the use of PFP to remove non-speech artifacts which result from the termination (disconnection) of a land line telephone call. Disconnection of a telephone call over a land line often results in a spike in the telephone signal, producing an audible ‘click’.

Such a spike is clearly visible in the utterance shown in Figure 4.9 (a). The spectrogram for this utterance is shown in Figure 4.9 (b) and the frame power in Figure 4.9 (c).

The morphologically-filtered frame power \mathbf{p}_{morph} for the call disconnect spike removal

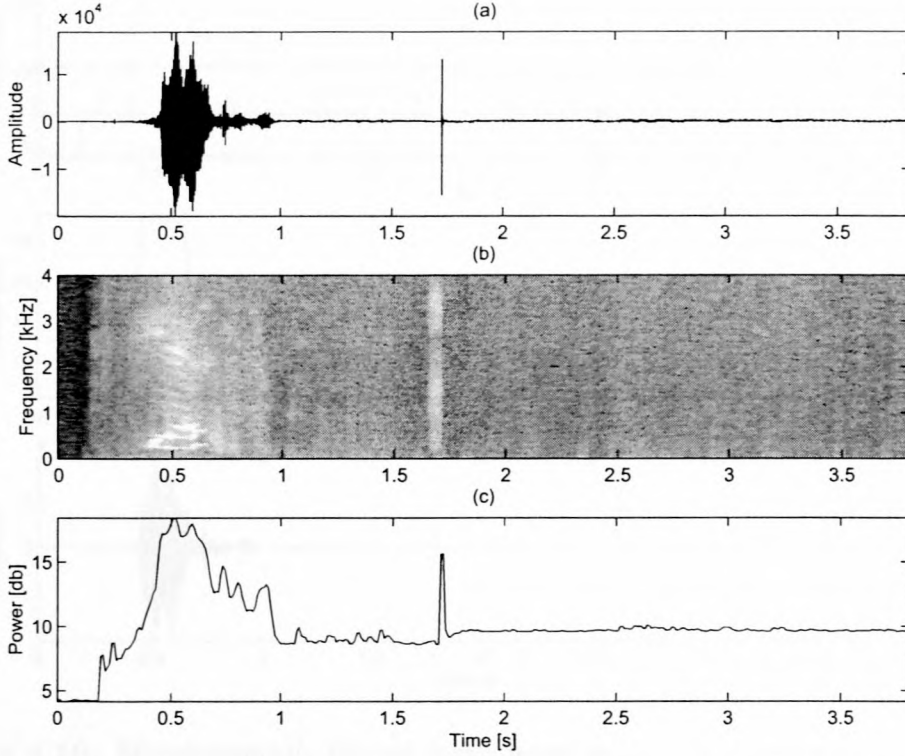


Figure 4.9: Speech signal (a), spectrogram (b) and frame power \mathbf{p}_{frm} (c) for the example utterance used to illustrate the removal of call disconnect spikes using the PFP.

example utterance is shown in Figure 4.10 (a). The percentile levels and power level thresholds used for percentile normalisation are also shown. The percentile-normalised frame power \mathbf{p}_{p2} is shown in Figure 4.10 (b). The speech signal following application of the PFP is shown in Figure 4.10 (c).

4.2 Speech features

The speech features are used to parameterise the input speech of the speech recognition system. For sensible and useful results the features must be chosen to reflect those characteristics of the input speech the experimenter wishes to model. Typically these characteristics will be frequency domain (spectral) characteristics, with higher-level linguistic knowledge built into the speech modelling stage.

Prior to the computation of the speech features, the original speech signal is divided into a number of overlapping frames to which are applied a windowing function such as a Hamming window [33]. The frames have a predetermined length (frame length) and overlap (frame overlap or skip). For this thesis, a frame length of 20ms and frame skip of

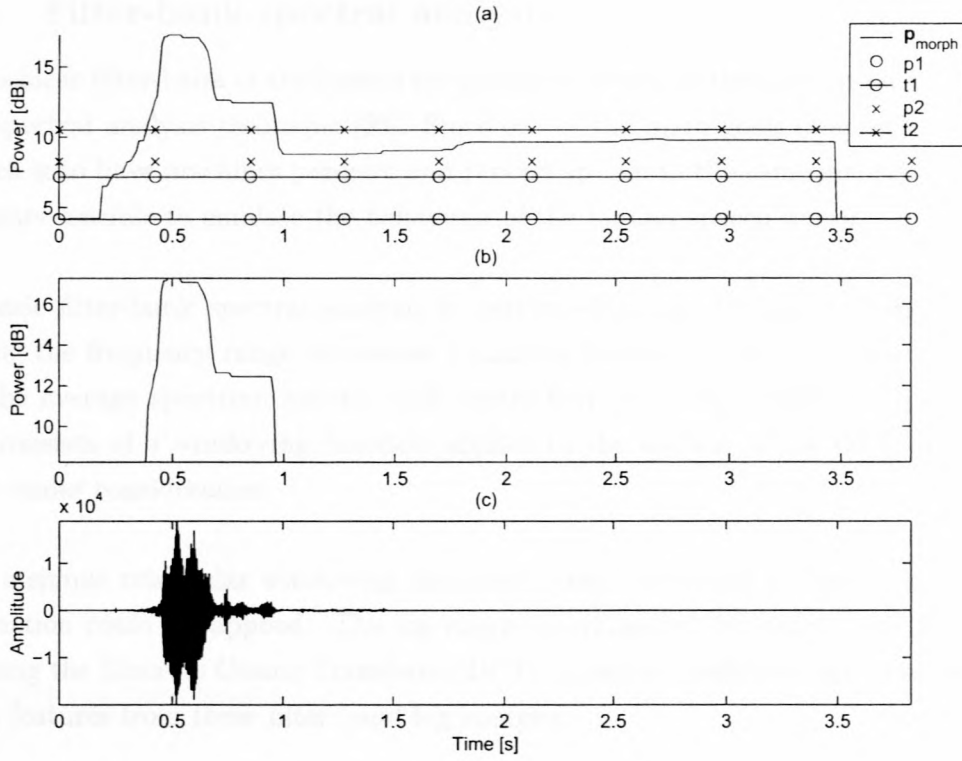


Figure 4.10: Morphologically filtered frame power p_{morph} (a), percentile-normalised frame power p_{p2} (b) and modified speech signal (c) for the example utterance used to illustrate the removal of call disconnect spikes using the PFP.

10ms was used.

The speech features are extracted on a frame-by-frame basis to produce a matrix of speech features for each input speech signal. A given row index of the feature matrix represent the speech features of the speech frame with the same index. A row of the feature matrix is often referred to as a feature vector.

The following speech features were investigated in this thesis:

- Features obtained from filter-bank spectral analysis of speech (MFCCs and others).
- Formant frequencies- and bandwidths.
- Combination MFCC and formant frequencies.
- Perceptual Linear Prediction (PLP) features.

A short description of each speech feature representation will be given next.

4.2.1 Filter-bank spectral analysis

The cochlear filter-bank of the human ear probably served as the inspiration for the filter-bank spectral analysis technique [20]. Since one of the main goals of speech recognition research is to have machines perceive and process speech in the same manner as humans, it appears sensible to emulate the behaviour of the human speech sensor.

The basic filter-bank spectral analysis is performed using overlapping bandpass filters covering the frequency range of interest ('analysis frequency range'). These filters compute the average spectrum around each centre frequency [18]. Each bandpass filter in effect consists of a windowing function applied to the section of the DFT of the input speech under consideration.

Often a simple triangular windowing function is used; although in theory, any windowing function could be applied. The log-energy is computed for each of the filter bands and using the Discrete Cosine Transform (DCT), cepstral coefficients are obtained as the speech features from these filter band log energies:

$$c(n) = \sum_{m=0}^{M-1} S(m) \cos \left[\frac{\pi n(m + \frac{1}{2})}{M} \right], \quad (4.7)$$

where $c(n)$ is the n 'th cepstral coefficient, M the number of filter bands and $S(m)$ is the log-energy output of the m 'th filter band which is given by:

$$S(m) = \ln \left[\sum_{k=0}^{N-1} |X(k)|^2 H_m(k) \right]. \quad (4.8)$$

N is the size of the DFT used, X is the DFT of the input signal and H_m is the windowing function for filter band m [18]. Figure 4.11 illustrates the filter-bank spectral analysis conceptually (after Arslan and Hansen). Two versions of the filter-bank spectral analysis method were utilised in this thesis.

The first method used a filter-bank with filter centre frequencies spaced according to the Mel frequency scale, whilst the second method used filter centre frequencies spaced according to the scheme proposed by Arslan and Hansen [4]. The features produced by the first method are commonly known as Mel Frequency Cepstral Coefficients (MFCCs).

Mel Frequency Cepstral Coefficients (MFCCs)

Mel Frequency Cepstral Coefficients are computed using filter bands with centre frequencies spaced equally in the Mel frequency domain. The Mel frequency scale is perceptually

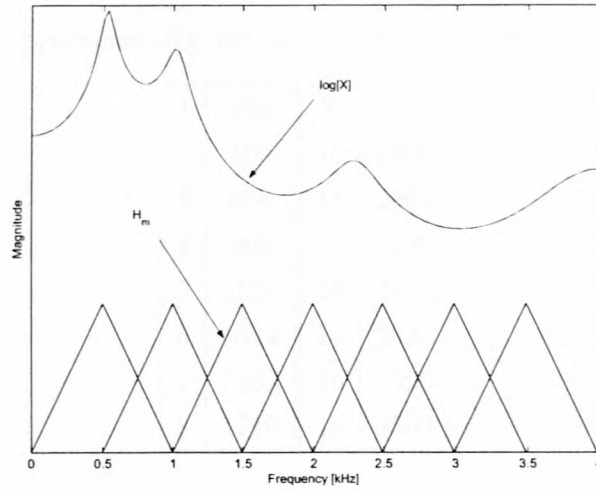


Figure 4.11: *Conceptual representation of filter-bank spectral analysis (after Arslan and Hansen).*

motivated and the conversion between linear frequency and Mel frequency is given by [18, 9]:

$$f_{mel}(f) = 1125 \ln \left[1 + \frac{f}{700} \right], \quad (4.9)$$

where f_{mel} represents the Mel-scale frequency and f represents the linear-scale frequency in Hertz. MFCCs find widespread application in modern speech recognition systems.

Filter-bank configuration of Arslan and Hansen

Arslan and Hansen deemed MFCCs to be unsuitable for accent- and language identification experiments [4]. Arslan and Hansen did not question the filter bank spectral analysis approach, but rather the placement of the filter band centre frequencies.

The typical filter-bank spectral analysis approach utilises filter band centre frequencies spaced evenly over the analysis frequency range. Arslan and Hansen proposed a filter-bank configuration consisting of sixteen overlapping filter bands, with a filter band centre frequency spacing which emphasises frequencies in the range 1.5-2.0kHz.

The authors derived this frequency scale based on a number of experiments to determine the discriminatory capabilities of different resonant frequencies. Table 4.1 lists the centre frequencies used in the filter-bank configuration of Arslan and Hansen.

Table 4.1: Centre frequencies [Hz] for filter-bank configuration of Arslan and Hansen.

1	250	9	1813
2	469	10	1906
3	688	11	2063
4	906	12	2250
5	1125	13	2438
6	1344	14	2625
7	1563	15	2906
8	1719	16	3500

4.2.2 Formant frequencies- and bandwidths

Formant frequencies are the centre frequencies of resonances in the speech spectrum, with these resonances indicating regions of emphasis in the speech. The resonances are the result of acoustical cavities formed in the vocal tract and dictate the speech frequency spectrum envelope [9]. The formant bandwidths are the -3dB peak widths of these resonances. Another interpretation of the formant frequencies is [20]:

“... they represent the frequencies that pass the most acoustic energy from the source to the output.”

Using an LP (Linear Prediction) filter representation of the input speech, one is able to obtain a smoothed spectrum envelope from which it is easier to locate the formants by using peak picking. A more accurate method of obtaining the formant frequencies involves solving the LP filter polynomial roots [5].

The LP filter polynomial is given by [5]:

$$V(z) = \prod_{i=1}^n (1 - z_i)(1 - z_i^*) / \prod_{i=1}^n (z - z_i)(z - z_i^*), \quad (4.10)$$

where z_i represents the i 'th complex conjugate pole pair. The roots of the LP filter polynomial take the form [5]:

$$z_i = r_i e^{2\pi j f_i T}, \quad (4.11)$$

with r_i the radius for the i 'th root-pair, f_i the formant frequency for the i 'th root-pair and T the sampling interval. The formant frequency f_i and bandwidth b_i are related to the i 'th polynomial root by [5]

$$f_i = (1/2\pi T) \text{Im}(\ln z_i), \quad (4.12)$$

and

$$b_i = (1/\pi T) \operatorname{Re}(\frac{1}{\ln z_i}). \quad (4.13)$$

Often only the first three formants are used in speech recognition applications [9]. It is possible to use knowledge of typical formant frequency- and bandwidth ranges to constrain the frequency- and bandwidth ranges on a per-formant basis. The paper by Fung and Kat presents an accent identification system utilising formant frequencies [21].

4.2.3 Combining MFCCs and formant frequencies

The combination of MFCCs, formant frequencies and formant bandwidths to form a new feature set was also investigated (section 6.2). The new feature set was formed by appending one feature set to the other on a per-vector basis. Experimental investigation showed that MFCCs of dimension nine yielded satisfactory classification results. LP filter analysis of order eight was used to obtain the formant frequencies- and bandwidths.

4.2.4 Perceptual Linear Prediction (PLP) features

Perceptual Linear Prediction (PLP) extends the Linear Prediction (LP) speech analysis technique in order to obtain a speech parameterisation which is more consistent with human hearing [15].

LP analysis results in a speech spectrum approximation with constant spectral resolution at all frequencies. Empirical studies have shown this approximation to be inconsistent with human speech perception. PLP extends LP speech analysis to take into account the following characteristics of human speech perception:

- The spectral resolution of human speech perception decreases with frequency after 800 Hz.
- The human ear is especially sensitive to frequencies around 3,5 KHz.
- Frequencies with higher power tend to mask other frequencies in the same *critical band* (a frequency range in psychophysics experiments).

The non-linear spectral resolution of human speech perception is characterised by the *Bark* frequency scale. The Bark frequency scale is given by [15]:

$$f_{bark} = 6 \ln(f/1200\pi + \sqrt{(f/1200\pi)^2 + 1}), \quad (4.14)$$

where f_{bark} is the frequency in Bark, and f is the frequency in Hz. The audible spectrum is covered by the range 0-24 Bark, with one Bark covering one critical band.

The spectral modifications of the PLP procedure consist of the following:

- Windowing the time-domain speech signal with a Hamming window.
- Computing the short-term power spectrum from the FFT of the windowed time-domain speech signal.
- Converting the linear frequency (Hz) scale of the short-term power spectrum to Bark frequency scale, using the frequency conversion of Equation 4.14.
- Convolution of the short-term power spectrum (Bark frequency scale) with ‘critical-band filters’. The smoothed short-term power spectrum obtained in this manner has a spectral resolution which is constant on the Bark scale and therefore more consistent with human speech perception.
- Down-sampling of the smoothed short-term power spectrum by sampling approximately every one Bark over the range 0-17 Bark.
- Applying equal-loudness preemphasis to model the frequency sensitivity of human hearing. The equal-loudness preemphasis transform is given by:

$$E(f) = \frac{(f^2 + 56,8 \times 10^6)f^4}{(f^2 + 6,3 \times 10^6)^2(f^2 + 0,38 \times 10^9)(f^6 + 9,58 \times 10^{26})}, \quad (4.15)$$

where f is the frequency in Hz and $E(f)$ is the gain as a function of frequency.

- Application of the intensity-loudness power law (or ‘cube-root amplitude compression’ in [15]). The intensity-loudness power law describes the non-linear relationship between the intensity of a sound and its perceived loudness. The intensity-loudness power law is given by:

$$L(f) = \sqrt[3]{I(f)}, \quad (4.16)$$

with $L(f)$ the perceived loudness at a given frequency f in Hz and $I(f)$ the intensity at the same frequency.

Standard LP analysis is then performed as follows:

- The inverse FFT (IFFT) of the modified short-term power spectrum is computed to obtain an autocorrelation sequence.
- The PLP parameters (modified LP parameters) are obtained from the autocorrelation sequence by solving the normal LP equations.
- The PLP parameters may then be converted to PLP cepstral coefficients if required.

4.3 Feature normalisation

Feature normalisation transforms the speech features in a number of ways which are designed to enhance characteristics beneficial to subsequent pattern modelling and to suppress the rest.

Feature normalisation is to features what preprocessing is to speech. It is not always that clear where feature normalisation ends and where pattern modelling begins. Next we present a number of feature normalisation techniques which were utilised in this thesis.

4.3.1 Feature scaling

Feature scaling (FS) is utilised to prevent numerical problems with feature vectors. It involves multiplying the feature vector elements with a scaling factor, using a separate scaling factor for each dimension (column) of the feature matrix. This operation may be represented in terms of matrix multiplication:

$$\hat{\mathbf{X}} = \mathbf{XS}, \quad (4.17)$$

$\hat{\mathbf{X}}$ is the transformed feature matrix, \mathbf{X} is the original feature matrix and \mathbf{S} is the diagonal scaling matrix containing the scaling coefficients on its main diagonal.

4.3.2 Feature Mean Subtraction

Feature Mean Subtraction (FMS) is used to enhance the speech recognition system's robustness against different microphone or channel transfer functions [18]. FMS consists of subtracting from each feature vector dimension the mean value for that feature vector dimension, as computed over a number of consecutive feature vectors:

$$\hat{\mathbf{x}}_i = \mathbf{x}_i - \bar{\mathbf{x}}_i, \quad (4.18)$$

with $\hat{\mathbf{x}}_i$ the normalised i 'th column of the feature matrix, \mathbf{x}_i the original column vector and $\bar{\mathbf{x}}_i$ the mean of this column vector.

Microphone transfer function robustness is of great importance when one deals with telephone speech; since unless every caller uses the same handset, one would be faced with a different microphone transfer function for every different handset used in the speech corpus collection and effectively a different channel for each call.

4.3.3 Velocity and acceleration coefficients

Standard speech feature representations such as MFCCs do not contain any information regarding the dynamic behaviour of the data being modelled. The change of feature co-

efficients over time can be presented with velocity or *delta*(Δ) coefficients. Δ coefficients are computed per feature vector dimension.

The velocity coefficients for the i 'th feature matrix column are given by:

$$\Delta x_i(j) = -2x_i(j-2) - x_i(j-1) + x_i(j+1) + 2x_i(j+2) \quad 2 \leq j \leq K-2, \quad (4.19)$$

with $x_i(j)$ the j 'th element of feature matrix column i , $\Delta x_i(j)$ the corresponding *delta* coefficient and K the number of rows in the feature matrix (elements in feature matrix columns). The first two elements as well as the last two elements of the feature matrix column under consideration are computed as such:

$$\Delta x_i(j) = \begin{cases} 2x_i(j+1) - 2x_i(0) & 0 \leq j \leq 1 \\ 2x_i(K-1) - 2x_i(j-1) & K-2 \leq j \leq K-1 \end{cases} \quad (4.20)$$

The acceleration or *deltadelta*($\Delta\Delta$) coefficients are obtained by applying Equation 4.19 to the velocity coefficients. The Δ coefficients presented here differ somewhat in terms of their computation from those presented in [18], but both versions of the Δ coefficients attempt to augment the static features with features describing the dynamic behaviour of the speech.

4.3.4 Karhunen-Loeve Transform (KLT) [10, 22]

The Karhunen-Loeve Transform (KLT), also known as Principal Component Analysis (PCA), is a data analysis technique often employed in pattern recognition. In speech recognition applications, the KLT is used as a feature normalisation tool.

Feature normalisation through the KLT is used to decorrelate feature matrix dimensions and to discard those dimensions which provide redundant information regarding the feature space in question. This feature dimension reduction is attained with a minimal loss in representation accuracy. The KLT may be seen as a method for lossy feature matrix compression.

The decorrelation of feature matrix dimensions benefit some speech models which, by their very nature, rely on the feature space dimensions being uncorrelated. Reduction of feature matrix dimension lessens the computational burden and reduces information entropy. The KLT is especially useful if a number of cascaded feature normalisation procedures are carried out which result in an increase in feature matrix dimensionality.

The KLT is based upon the “principal axis theorem” of linear algebra [24]. The goal

of the KLT is to find an orthogonal transformation matrix P which results in the transformation of a feature matrix from a feature space where the feature matrix dimensions are correlated, to another feature space where the feature matrix dimensions are uncorrelated:

$$\mathbf{Y} = \mathbf{P}^T \mathbf{X}, \quad (4.21)$$

with \mathbf{Y} the feature matrix following application of the KLT, \mathbf{P}^T the transposed transformation matrix and \mathbf{X} the feature matrix prior to application of the KLT. See Figure 4.12 for a conceptual representation of this transformation. The covariance matrix \mathbf{D} of \mathbf{Y} is related to the covariance matrix $\mathbf{\Sigma}$ of \mathbf{X} as follows:

$$\mathbf{D} = \mathbf{P}^T \mathbf{\Sigma} \mathbf{P}. \quad (4.22)$$

\mathbf{D} is a diagonal matrix with the eigenvalues of $\mathbf{\Sigma}$ in descending order on its main diagonal, with \mathbf{P} containing the orthonormal eigenvectors of $\mathbf{\Sigma}$. These eigenvectors are shown as \mathbf{x}_1 and \mathbf{x}_2 in the conceptual representation of Figure 4.12. The orthonormal eigenvectors form the basis vectors of a new feature space in which the feature space dimensions are uncorrelated. The basis vectors of the corresponding feature space is shown in Figure 4.12 as \mathbf{y}_1 and \mathbf{y}_2 .

The eigenvalues contained in \mathbf{D} indicate the relative contribution of each feature vector dimension in the transformed feature space of \mathbf{Y} . Eigenvalues in \mathbf{D} which are small relative to the sum of the eigenvalues in \mathbf{D} , indicate feature vector dimensions in \mathbf{Y} which do not contribute much to the description of the data contained in \mathbf{Y} and which may therefore be discarded from \mathbf{Y} .

For practical speech recognition systems, one would typically specify the number of feature vector dimensions to retain following application of the KLT. One could also specify a percentage to indicate how much significant information of \mathbf{Y} should be retained following the discarding of insignificant feature vector dimensions:

$$\%_{ret} = \frac{\sum_{k=1}^R \alpha_k}{trace(\mathbf{D})}, \quad (4.23)$$

where $\%_{ret}$ is the percentage of the original information content to retain, the α_k are the R retained eigenvalues of \mathbf{D} and $trace(\mathbf{D})$ is the sum of the diagonal elements of \mathbf{D} . R is less than or equal to the number of diagonal elements in \mathbf{D} .

4.3.5 Feature Frame Expansion (FFE)

Feature Frame Expansion (FFE) normalisation is used to expand feature space dimensionality artificially. The basic normalisation process consists of framing the feature matrix

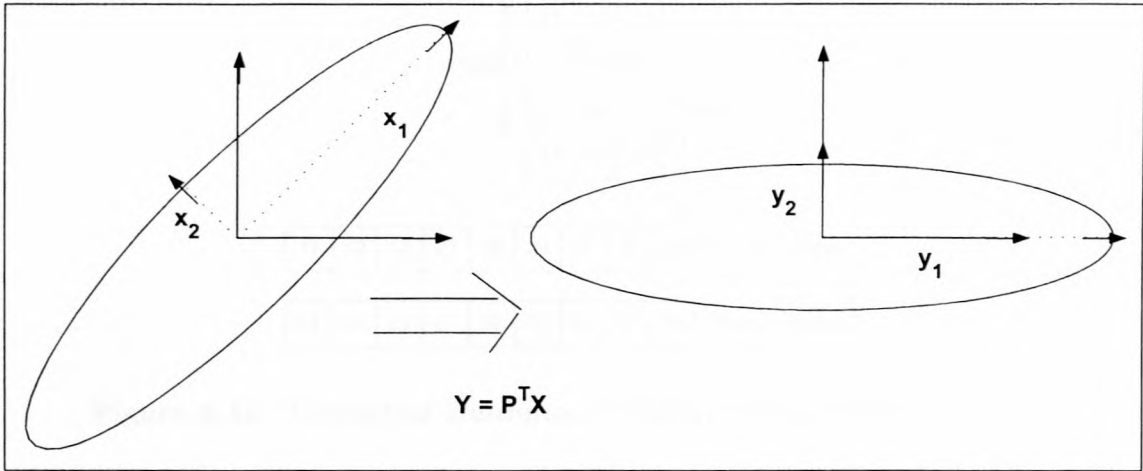


Figure 4.12: *Conceptual representation of the KLT.*

along its rows (feature vectors) and appending a pre-specified number (‘frame length’) of the framed feature vectors column-wise to the original set of feature vectors. This creates new feature vectors with a higher number of columns and hence higher dimensionality. The frame skip is specified in terms of the number of feature vectors (rows) in the original feature matrix to advance between successive framing operations.

Figure 4.13 shows the first four rows of a three-dimensional artificial feature matrix. The numerical values shown were chosen to aid in the discussion. Suppose this feature matrix is to be normalised using FFE normalisation, with a frame length of three feature vectors and a frame skip of one feature vector. Given the number of feature vectors shown in Figure 4.13 and the parameters of the FFE normaliser, this operation should yield two new feature vectors of dimensionality nine. The frames from which the new feature vectors are to be created, are labelled ‘Frame 1’ and ‘Frame 2’ in Figure 4.13. The elements of the new nine-dimensional feature vectors are also shown in Figure 4.13. The new feature vectors are labelled ‘Feature vector 1’ and ‘Feature Vector 2’ in Figure 4.13.

4.3.6 Feature Cross-term Expansion (FCE)

The FCE normaliser was inspired by the Volterra expansion and the application thereof to nonlinear filters [37]. The FCE normaliser appends to an input feature vector its cross-term expansion up to a predetermined polynomial power. For example, the second-order FCE of a two-dimensional feature vector with elements x and y is given by:

$$[x \ y] \Rightarrow [xy \ x^2 \ y^2]. \quad (4.24)$$

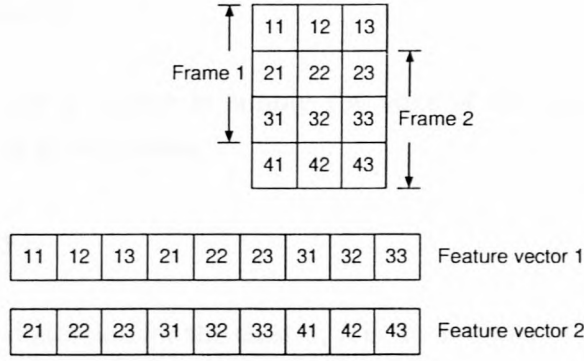


Figure 4.13: *Conceptual illustration of Feature Frame Expansion (FFE).*

This results in an expanded feature space. KLT normalisation may be used after the FCE normalisation to reduce the dimension of the expanded feature space whilst retaining the most significant feature space dimensions.

4.4 Vector Quantisation (VQ)

Vector Quantisation (VQ) partitions the feature space automatically into a predefined number of clusters, where each cluster is represented by the cluster mean vector or codebook vector. The key component of VQ is a distortion measurement which measures the quantisation error which results from approximating a feature vector \mathbf{x} by a codebook vector \mathbf{y} .

The distortion (quantisation error) as a result of approximating a feature vector \mathbf{x} by a codebook vector \mathbf{y} is given by [26]:

$$d(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^T (\mathbf{x} - \mathbf{y}) = \sum_{k=1}^N (x_k - y_k)^2, \quad (4.25)$$

where $d(\mathbf{x}, \mathbf{y})$ is the distortion for a feature vector \mathbf{x} relative to the codebook vector \mathbf{y} and N is the dimension of the feature vectors. The feature vector element for dimension k is denoted by x_k and the codebook vector element for dimension k is denoted by y_k .

The VQ for this thesis was performed using non-uniform binary-split to obtain the initial codebook centroids followed by refinement using the K-means algorithm. Non-uniform binary-split VQ begins by selecting at random two feature vectors from the set of feature vectors which are to be clustered. The two feature vectors selected in this manner are chosen to be the first two codebook vectors. The cluster having the greatest total distortion will be split again. This process is repeated until the required number of codebook

vectors has been produced.

The total distortion for a cluster is simply the sum of the distortions for all feature vectors assigned to the given cluster:

$$d_{tot} = \sum_{l=1}^L d(\mathbf{x}_l, \mathbf{y}), \quad (4.26)$$

where d_{tot} is the total distortion for the cluster, L is the number of feature vectors assigned to the cluster, \mathbf{y} is the codebook vector for the given cluster and $d(\mathbf{x}_l, \mathbf{y})$ is the distortion for feature vector \mathbf{x}_l relative to the codebook vector \mathbf{y} .

K-means clustering is used to refine the initial clustering obtained from the non-uniform binary-split. It is an iterative clustering technique which computes the Euclidean distance for each of the feature vectors relative to the codebook vectors. Each feature vector is then mapped to the cluster whose centroid is nearest in the Euclidean sense. After the feature vector remapping is complete, the cluster centroids are recomputed. This process is repeated until the overall distortion drops below a pre-defined threshold [20].

Figure 4.14 shows a generated two-dimensional feature space, which is to be Vector Quantised into a codebook consisting of three codebook vectors (cluster centroids), after the application of uniform binary-split. Figure 4.15 shows the final clustering obtained. In both Figure 4.14 and Figure 4.15, the codebook centroids are indicated by solid markers.

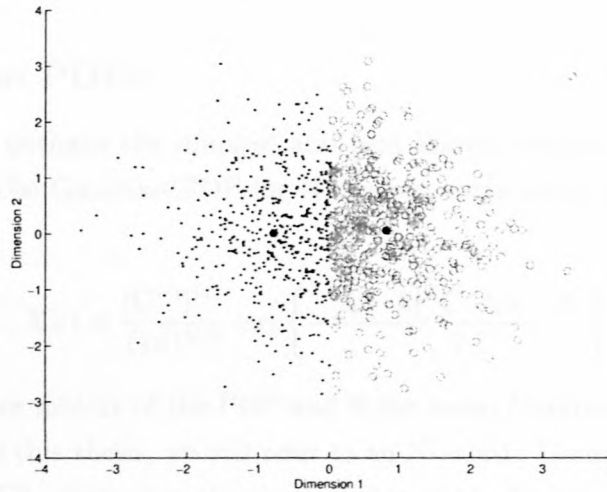


Figure 4.14: Initial clustering of an artificial feature space using binary split Vector Quantisation (VQ).

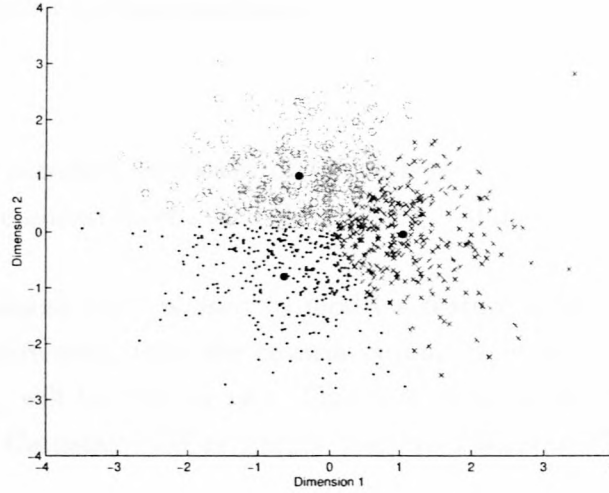


Figure 4.15: Final clustering for the artificial feature space of Figure 4.14, following application of the K-means algorithm.

4.5 Speech modelling

In this section a brief review of the statistic pattern modelling techniques employed in this thesis will be presented. These include:

- Gaussian Probability Density Functions (PDFs, subsection 4.5.1).
- Gaussian Mixture Probability Density Functions (GMMs, subsection 4.5.2).
- Hidden Markov Models (HMMs, subsection 4.5.3).

4.5.1 Gaussian PDFs

Gaussian PDFs are perhaps the simplest and best known probabilistic models used for speech modelling. The Gaussian PDF for N random variables is given by the following equation [32]:

$$f_{X_1, \dots, X_N}(X_1, \dots, X_N) = \frac{|\mathbf{C}^{-1}|^{1/2}}{(2\pi)^{N/2}} \exp\left\{-\frac{[\mathbf{x} - \bar{\mathbf{x}}]^T \mathbf{C}^{-1} [\mathbf{x} - \bar{\mathbf{x}}]}{2}\right\}, \quad (4.27)$$

with \mathbf{C} the covariance matrix of the PDF and $\bar{\mathbf{x}}$ the mean (centroid) vector of the PDF. For the remainder of this thesis, we will refer to an N -variate Gaussian PDF by the compact notation $N(\bar{\mathbf{x}}, \mathbf{C})$, where $\bar{\mathbf{x}}$ is the mean vector of the N -variate Gaussian PDF and \mathbf{C} its covariance matrix.

The correlation coefficient (ρ) of any two random variables forming part of an N -variate

Gaussian PDF is related to their covariance C_{ij} as follows [32]:

$$\rho = \frac{C_{ij}}{\sigma_i \sigma_j}, \quad (4.28)$$

with σ_i and σ_j the standard deviation of the respective random variables. If the two random variables are uncorrelated, then their corresponding covariance C_{ij} will be zero.

If an N-variate Gaussian PDF is used to model a feature space whose dimensions are assumed to be uncorrelated, then the covariance matrix of this PDF will be diagonal. For such a PDF, C_{ij} will be zero for $i \neq j$. This type of Gaussian PDF is referred to as a diagonal covariance Gaussian PDF or simply diagonal Gaussian PDF.

4.5.2 Gaussian Mixture PDFs (GMMs)

Gaussian Mixture PDFs or Gaussian Mixture Models (GMMs) are capable of modelling a complex feature space which cannot be modelled with simple Gaussian PDFs. A GMM consists of the weighted combination of a number of Gaussian PDFs. The PDFs which comprise the GMM are sometimes referred to as the ‘component densities’ of the GMM.

For observed data \mathbf{y} , the GMM consisting of K Gaussian PDFs is given by:

$$GMM(\mathbf{y}) = \sum_{k=1}^K w_k N(\mathbf{y}|\bar{\mathbf{x}}_k, \mathbf{C}_k), \quad (4.29)$$

where w_k is the weighting applied to the k ’th mixture component and $N(\mathbf{y}|\bar{\mathbf{x}}_k, \mathbf{C}_k)$ represents the Gaussian PDF of mixture component k . The parameters of the GMM are estimated using an iterative technique called the Expectation Maximisation (EM) algorithm [9].

4.5.3 Hidden Markov Models (HMMs)

A Hidden Markov Model (HMM) is a stochastic state machine which extends the Markov chain concept from basic statistics [34]. It has a number of interconnected states and the transition from state i to state j is governed by the transition probability or weight a_{ij} . The state machine nature of the HMM leads to the ability to model data that varies with time, such as speech.

Each state of the HMM has an output PDF which determines the output of the HMM when it is in a given state. The output PDFs of the HMM are sometimes known as the emitting densities of the HMM. The emitting densities of the HMM states could basically consist of any type of PDF.

In this thesis the HMM emitting densities consisted of single diagonal covariance Gaussian PDFs, single full covariance Gaussian PDFs and diagonal covariance GMMs. In speech recognition applications, the HMM emitting densities typically model the spectral characteristics of the speech. Figure 4.16 shows the conceptual representation for part of a first-order HMM.

The parameters of the HMM are determined from training observation sequences using a form of the EM algorithm, the Baum-Welch algorithm. The Viterbi algorithm is used for classifying an input vector sequence with a given HMM. The Viterbi algorithm may also be utilised for estimation of the HMM parameters. Practical experience has shown that the Viterbi algorithm is computationally more efficient than the Baum-Welch algorithm, at a negligible decrease in HMM overall recognition accuracy.

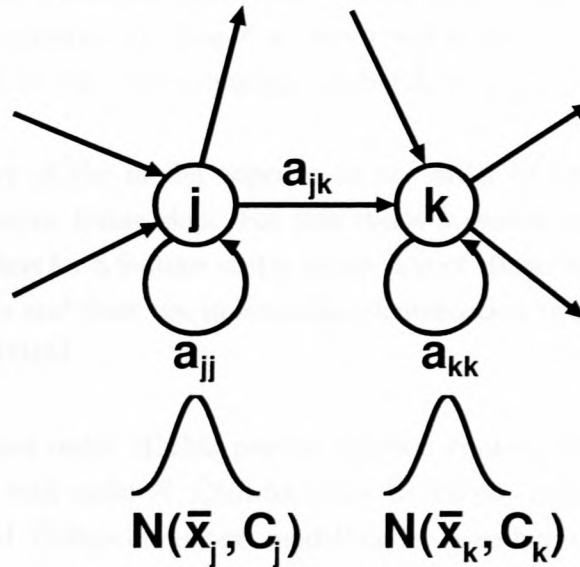


Figure 4.16: *Conceptual representation of a first-order HMM. Only two states of the HMM are shown.*

Initialising ergodic HMMs from GMMs

It is possible to represent a GMM with an ergodic HMM equivalent. Each component density of the source GMM will map to a unique state of the ergodic HMM. The transition weights for links terminating on a given HMM state, are set equal to the weight of the corresponding GMM component weight.

Initialising ergodic HMMs from GMMs may result in better initialisation of the HMM acoustic modelling component. This allows subsequent parameter estimation to focus on

fine-tuning the HMM transition weights and could lead to faster convergence of the HMM parameter estimation process.

Figure 4.17 illustrates the conversion of a GMM to an equivalent ergodic HMM. In this figure it will be seen that an additional weight, w_3 , is added to the HMM. This additional weight has no equivalent in the source GMM. It is added in order for the HMM to have a well-defined exit state. Since the HMM transition probabilities must sum to one, the original weights w_1 and w_2 need to be rescaled accordingly.

Higher-order HMMs

Higher-order HMMs simply extend first-order models by including more than one prior state in the computation of the state transition probabilities. For instance, a second-order HMM will have a state transition dependency on two prior states instead of the one of a first-order HMM. A transition to state k at time t will now depend on prior states i and j and will be denoted by the state transition probability a_{ijk} .

The modelling history of the model depends on the order of the model and the duration of the feature vector frame skip. For this thesis a second-order model would have duration history of 20ms for a feature vector frame skip of 10ms. Refer to [11] for a review of higher order HMMs and their use in language identification tasks. Figure 4.18 shows a partial second-order HMM.

As noted in [11], higher order HMMs provide implicit context modelling over the prior N states for a model with order N . Context refers to the co-occurrence of the categories modelled by the HMM. Sufficient context modelling for speech units like phonemes would require a model of unrealistically high order. This is again due to the fact that the context memory of the model is determined by the product of the frame-skip and the order of the model.

The context-emphasised higher order models overcome this problem by making the state transition probabilities independent of the number of times a prior state occurred, i.e. only the sequence of prior states matters. For the context-emphasised models a plus (+) superscript is used to indicate one or more occurrence of a given state. Figure 4.19 shows a third-order context-emphasised left-to-right HMM with single state skip.

A duration-emphasised HMM explicitly models the duration (state occupation time) for the categories being modelled by the HMM. This type of model will probably be better suited to the modelling of specific speech units, than modelling the general spectral and

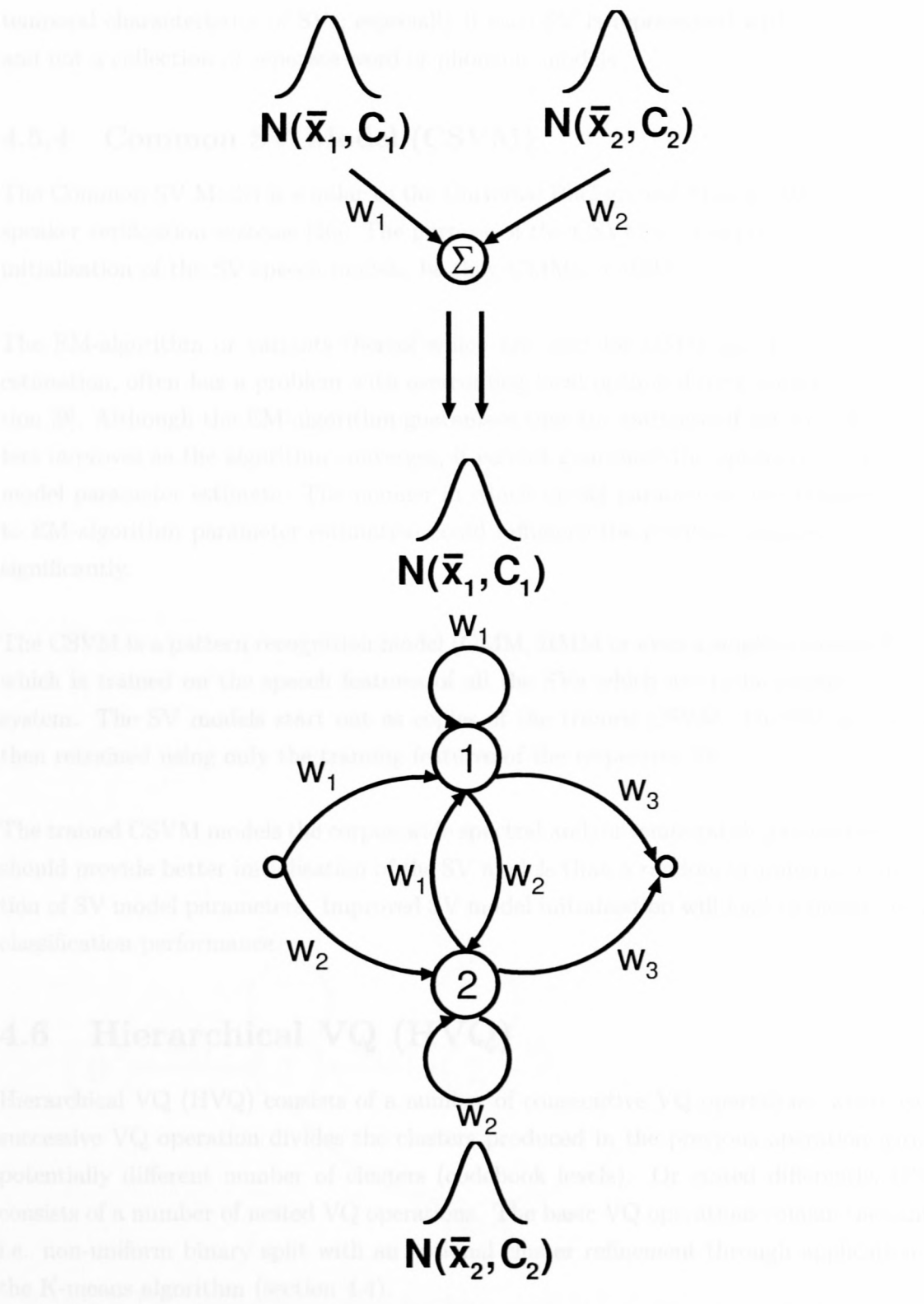


Figure 4.17: Ergodic HMM initialised from a GMM. Transition weights for links terminating on a given HMM state are set equal to the weight of the corresponding GMM component weight. The corresponding GMM component density is utilised as HMM emitting density. The additional weight w_3 added to the HMM ensures that the HMM has a well-defined exit state.

temporal characteristics of SVs; especially if each SV is represented with a single HMM and not a collection of separate word or phoneme models.

4.5.4 Common SV Model (CSVM)

The Common SV Model is similar to the Universal Background Model (UBM) utilised in speaker verification systems [36]. The purpose of the CSVM is to improve the parameter initialisation of the SV speech models, be they GMMs or HMMs.

The EM-algorithm or variants thereof which are used for GMM and HMM parameter estimation, often has a problem with overcoming local optima during parameter estimation [9]. Although the EM-algorithm guarantees that the estimate of the model parameters improves as the algorithm converges, it cannot guarantee the optimality of the final model parameter estimate. The manner in which model parameters are initialised prior to EM-algorithm parameter estimation could influence the eventual parameter estimate significantly.

The CSVM is a pattern recognition model (GMM, HMM or even a single Gaussian PDF), which is trained on the speech features of all the SVs which are to be classified by the system. The SV models start out as copies of the trained CSVM. The SV models are then retrained using only the training features of the respective SV.

The trained CSVM models the corpus-wide spectral and/or temporal characteristics. This should provide better initialisation of the SV models than a random or uniform initialisation of SV model parameters. Improved SV model initialisation will lead to better overall classification performance.

4.6 Hierarchical VQ (HVQ)

Hierarchical VQ (HVQ) consists of a number of consecutive VQ operations, where each successive VQ operation divides the clusters produced in the previous operation with a potentially different number of clusters (codebook levels). Or stated differently, HVQ consists of a number of nested VQ operations. The basic VQ operations remain the same, i.e. non-uniform binary split with an optional cluster refinement through application of the K-means algorithm (section 4.4).

The need for hierarchical VQ arose from experiments involving the use of GMMs as emitting PDFs for the HMMs used in this thesis. One could use GMMs with parameters initialised to arbitrary values, but this might lead to unsatisfactory parameter estimation

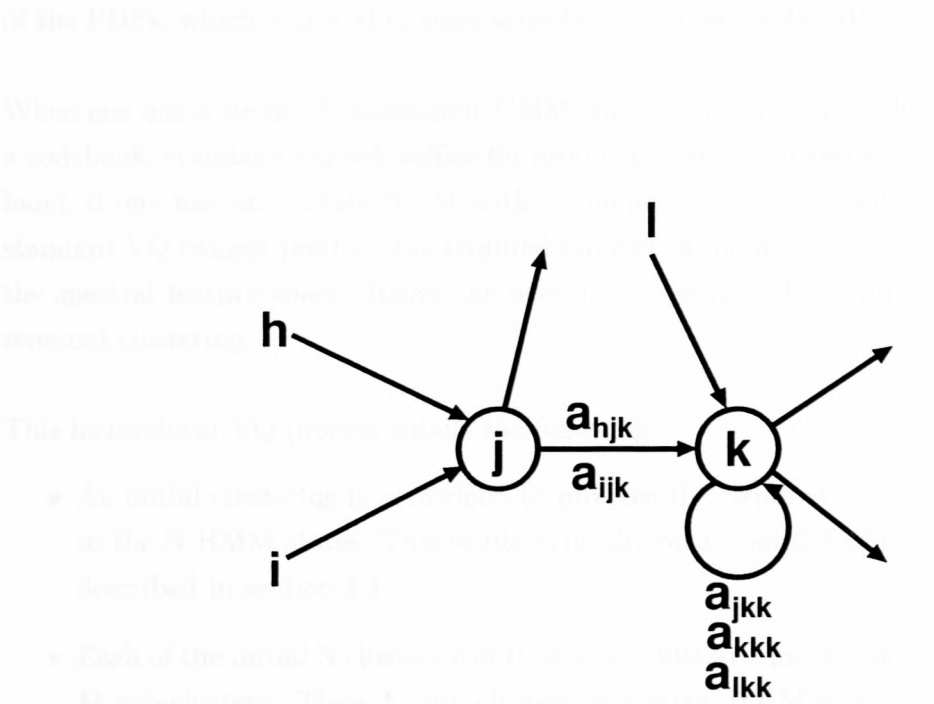


Figure 4.18: Partial second-order HMM (after Du Preez and Weber [12]).

4.7 Speech classification

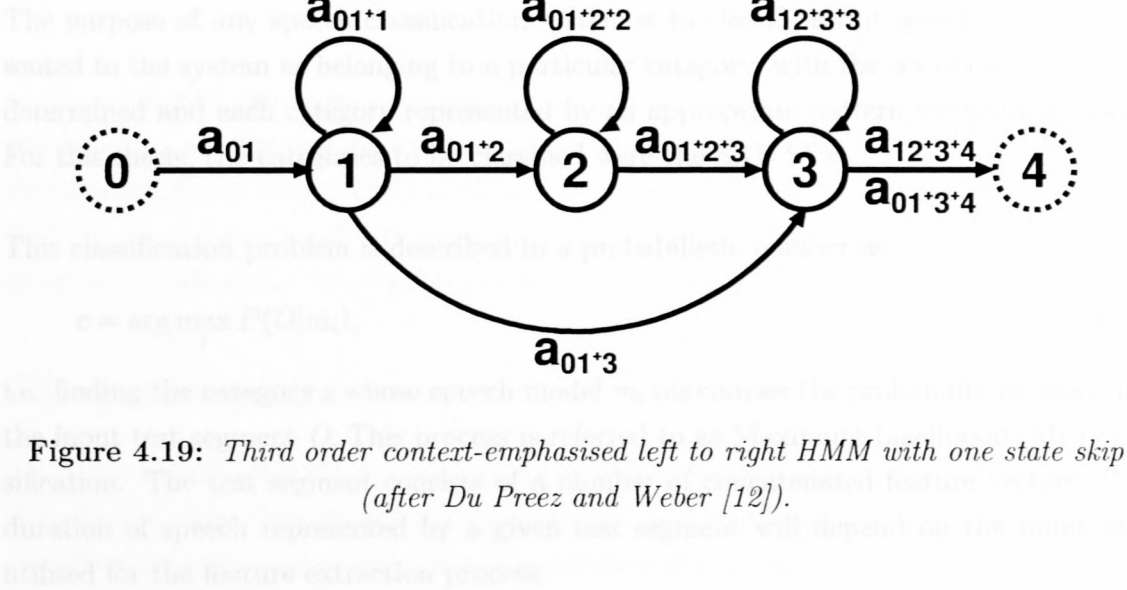


Figure 4.19: Third order context-emphasised left to right HMM with one state skip (after Du Preez and Weber [12]).

of the PDFs, which will lead to poor acoustic modelling by the HMMs.

When one has a single M -component GMM whose centroids one wishes to initialise from a codebook, standard VQ will suffice for producing the M clusters required. On the other hand, if one has an N -state HMM with N unique M -cluster GMMs as emitting PDFs, standard VQ cannot produce the required number of unique clusters automatically from the spectral feature space. Hence the need for a specialised VQ process to produce the required clustering.

This hierarchical VQ process entails the following:

- An initial clustering is performed to produce the required N -clusters corresponding to the N HMM states. This would typically be accomplished using the VQ process described in section 4.4.
- Each of the initial N clusters will then be re-clustered independent of each other into M sub-clusters. These M sub-clusters of a given HMM state are used to initialise the component PDF centroids of the GMM for that state.
- The nested VQ process may theoretically be repeated for as many levels of nesting required by the particular speech modelling approach utilised. For typical speech recognition configurations one should rarely need to use more than two levels of nested VQ.

4.7 Speech classification

The purpose of any speech classification system is to classify a test speech segment presented to the system as belonging to a particular category, with the set of categories predetermined and each category represented by an appropriate pattern recognition model. For this thesis, the categories to be classified were the SAE SVs.

This classification problem is described in a probabilistic manner as:

$$c = \arg \max_i P(O|m_i), \quad (4.30)$$

i.e. finding the category i whose speech model m_i maximises the probability of observing the input test segment O . This process is referred to as Maximum Likelihood (ML) classification. The test segment consists of a number of concatenated feature vectors. The duration of speech represented by a given test segment will depend on the frame skip utilised for the feature extraction process.

The classification strategy used in the thesis experiments was quite simple. Each block of feature vectors (or ‘test segment’) used in the classification process was scored against each SV model in turn. The variant classification decision was made for that model which produced the highest log-likelihood, i.e. the most likely variant.

In all of the classification experiments the category membership of each test segment was known at all times. Therefore it was a simple matter to determine whether a test segment was classified correctly or not.

A confusion matrix was used throughout the experiments to indicate the correct and incorrect classifications made. The rows of the matrix represent the actual or true categories and the columns of the matrix represent the confused categories. I.e. confusion matrix element (i,i) indicates the number of times category i’s test segments were correctly classified. An off-diagonal element (i,j) indicates the number of times the test segments of category i were mistakenly classified as belonging to category j. By dividing the number

Table 4.2: *Confusion matrix with corresponding classification accuracies and RER values for a typical classification experiment.*

		Confused categories					Accuracy	RER
		AE	BE	CE	EE	IE		
Actual categories	AE	199	7	5	28	17	$199/256 = 77,73\%$	22,27%
	BE	30	312	10	41	22	$312/415 = 75,18\%$	24,82%
	CE	16	7	239	18	11	$239/291 = 82,13\%$	17,87%
	EE	19	8	0	228	23	$228/278 = 82,01\%$	17,99%
	IE	7	3	2	31	246	$246/289 = 85,12\%$	14,88%

of correct classifications for a category by the number of test segments which occurred for the category, one obtains the classification accuracy obtained for the given category. The Recognition Error Rate (RER) is often used instead of the classification accuracy. The RER is simply the classification accuracy percentage subtracted from 100%. Table 4.2 shows a hypothetical confusion matrix for a classification experiment.

Chapter summary

The basic speech recognition theoretical concepts utilised in this thesis were presented in this chapter. The theoretical material presented in this chapter form the basis of the SV classification systems developed in the experimental investigation chapter (chapter 6).

A couple of signal processing- and pattern recognition techniques developed for this thesis were presented in this chapter. The CARP preprocessor was created to remove speech segments with constant amplitude (subsection 4.1.3). Speech segments of zero-amplitude and duration equal to or greater than the feature extraction frame length, may cause numerical instabilities during the feature extraction process. The CARP allowed for the automatic removal of speech files which contained no speech at all.

Another preprocessor, the PFP, was created to enhance system robustness against non-speech artifacts introduced during the speech recording stage. The algorithm of the PFP was discussed using an example utterance. Two additional examples illustrating the use of the PFP to remove non-speech artifacts were presented after the algorithm discussion (subsection 4.1.4).

A modification of the basic VQ process, HVQ was presented in this chapter (section 4.6). It was designed to obtain automatically the feature space partitioning required for use of GMMs as emitting PDFs in HMMs trained from spectral features only.

In the following chapter we discuss the pattern recognition configurations which were considered for SV classification. These configurations were based on the results of the literature study and some of the theoretical concepts presented in this chapter.

Chapter 5

Recognition configurations for SAE SV classification

Introduction

This chapter presents a number of speech recognition configurations which were considered for SAE SV classification. The literature synopsis of chapter 2 serves as background for this chapter, with specific reference to the literature synopsis summary in section 2.3. The theoretical concepts utilised in this chapter were described in chapter 4.

The layout for this chapter is as follow:

- Applicable speech recognition configurations (section 5.1).
- Speech recognition configurations selected (section 5.2).

5.1 Applicable speech recognition configurations

Based upon the literature synopsis, a number of speech recognition configurations were considered for classification of SAE SVs.

5.1.1 Phoneme recognition followed by N-gram modelling

The first proposed speech recognition configuration is one which is similar to the LID approach used by Zissman et al (subsection 2.2.2). As described before, this configuration utilises a combination of phoneme recognisers and N-gram language models trained from speech transcriptions.

The following disadvantages are foreseen for this type of configuration:

- The phoneme recognisers utilised in the system front-end are inherently inaccurate. Typical phoneme recogniser accuracies, obtained on corpora such as TIMIT [13], are in the 50-60% range.
- The phoneme recogniser front-end introduces a hard classification decision very early in the recognition process. Coupled with the accuracy (or lack thereof) of the phoneme recognisers, this could result in poor classification performance for the system in general.
- Speech transcriptions must be available for the corpus on which the phoneme recognisers are to be trained. Producing the speech transcriptions is a costly and time-consuming process, especially for the larger conversational speech corpora.

5.1.2 Gaussian Mixture Models (GMMs)

GMMs (subsection 4.5.2) may be utilised for the SV modelling if the SVs are to be distinguished by their spectral characteristics alone. In this instance it is assumed that SV temporal characteristics do not affect the classification process. The major advantage (and possible disadvantage) associated with a GMM-based SV classification configuration is the simplicity of the modelling technique employed. The GMMs may be trained with or without speech transcriptions.

GMMs trained with speech transcriptions

Each component probability density function (PDF) of a SV GMM could model a phoneme, with the allocation of training features to specific GMM components determined from phonetic speech transcriptions. The GMM components representing individual phonemes are still trained on spectral speech features.

GMMs trained with spectral features only

If speech transcriptions are not to be used, the GMM component clustering could be obtained automatically from the spectral feature space using vector quantisation (VQ). The GMM component densities then no longer map to known speech units or categories. The underlying unknown categories will be referred to as ‘pseudo-phonemes’.

Improving GMM initial conditions via a Common SV Model (CSVM)

The EM (Expectation Maximisation) algorithm is used for GMM parameter estimation [9]. Although the EM algorithm guarantees that the estimate of the model parameters

improves as the algorithm converges, it cannot guarantee the optimality of the the final model parameter estimate. Model initialisation prior to EM-algorithm parameter estimation could influence the eventual parameter estimate and classification performance significantly.

Typical GMM parameter initialisation entails the following:

- The GMM component PDF centroids are initialised via VQ codebook.
- Covariance matrices are initialised to sensible arbitrary values.
- The GMM component weights are initialised to equal values (equal component weighting initially), or if the prior distribution of the modelled categories is known, the GMM component weights may be set accordingly.

One method which might provide better initialisation of the SV GMMs, is the use of a CSVM (Common SV Model). This model is similar to the UBM (Universal Background Model) used in speaker verification systems [36]. The CSVM is a model which is trained on the features of all the SVs being modelled. Each of the SV GMMs is an exact copy of the trained CSVM and is then retrained on the features of the specific SV. The VQ codebook used to initialise the CSVM component density centroids is computed from the features of all the SVs pooled together.

5.1.3 N-th order ergodic HMMs

Another possible approach for SAE SV classification involves the use of N-th order, including first-order, ergodic (fully-connected) HMMs. These HMMs could be trained with or without speech transcriptions. Ergodic HMM link structures are chosen to model the SVs, since we have no prior knowledge regarding the temporal characteristics of the SVs. I.e. it is not known which HMM link structure will be appropriate for modelling the SVs.

HMMs trained with speech transcriptions

If phonetic speech transcriptions are used to train the HMMs with, then each state of the HMM will represent a phoneme. The number of states in the HMM will correspond to the number of phoneme categories contained in the speech transcriptions.

HMMs trained with spectral features only

If the HMMs are trained from spectral features only, then the states of the HMMs no longer map to known speech units. Again, the unknown speech categories modelled here will be referred to as ‘pseudo-phonemes’. When training the HMMs from spectral features

alone, VQ codebooks are used to initialise the centroids of the HMM emitting PDFs. The number of states (PDFs) is then simply the number of clusters used for VQ.

Standard VQ techniques will only provide a clustering of the feature space which is suitable for using a single PDF per HMM state as the emitting density. If one is to use mixture PDFs (such as GMMs) for the HMM emitting densities, then a multi-layered VQ approach is required to generate the codebooks required. Such a multi-layered VQ technique, known as Hierarchical VQ (HVQ), was implemented for this thesis and is explained in more detail in section 4.6.

Higher order HMMs

Different order HMM configurations may be utilised for the SV modelling. HMMs of order two and above increase the number of previous HMM states considered for each state transition. These higher order HMMs are initialised from a trained lower order HMM via the ORED (ORder rEDucing) algorithm [12]. The disadvantages associated with higher order HMMs include their sensitivity to data scarcity and increased computational requirements. The computational requirements issue is addressed by the FIT (Fast Incremental Training) algorithm, which enables efficient training of HMMs of arbitrary order [12].

Context- and duration emphasised HMMs

Besides investigating the role of HMM order on the classification performance, one might also consider the use of context- and duration-emphasised HMMs. Context-emphasised HMMs represent the co-occurrence of the modelled categories. Duration-emphasised HMMs model the duration of each category. Context- and duration emphasised HMMs allow for higher-level linguistic constraints to be imposed on the SV models.

Different methods of HMM parameter initialisation

Different methods for initialisation of the SV HMMs could also be investigated. HMMs trained on spectral features only, have their emitting density centroids initialised via VQ and their link structure weights initialised to fixed initial values. This method provides adequate initialisation of the spectral component of the HMM but not its temporal component (the link structure). At best one could initialise the transition probabilities of the HMM to values reflecting the prior distribution of the categories being modelled.

HMM parameters are estimated via variants of the EM-algorithm and the HMM parameter estimation process is therefore also affected by the local optima problem. A CSVM

HMM could be utilised to provide better initialisation of the SV HMMs. Another method of initialising the SV HMMs is via trained SV GMM models, since it is possible to represent a GMM with a first order ergodic HMM equivalent (subsection 4.5.3).

5.1.4 Hierarchical HMMs (HHMMs)

Another possible approach for SAE SV classification involves the use of an HHMM (Hierarchical HMM) structure. An HHMM is a multi-layered HMM where each layer of the HMM, except for the deepest (or bottom) layer, has an HMM embedded at each of its state nodes. The layer of the HHMM which is not contained in any of its other layers, will be called the ‘top-layer’. The layer of the HHMM which does not have any HMMs embedded in any of its states, ie. it is a standard single-layer HMM, will be called the ‘bottom-layer’.

A standard single-layer HMM can only provide implicit modelling of phonemes with each of its states. Using a two-layer HHMM, one could have phoneme modelling HMMs (three-state left-to-right HMMs typically) embedded at each state node of the top-layer HMM. The top-layer HMM structure will be initialised to random values or to values reflecting the prior distribution of each of the phoneme categories.

The key concept here is to model the interaction between the different phoneme categories. To enforce this concept, the phoneme HMMs will be trained and then have their parameters frozen prior to embedding them in the top-layer HHMM. Training of the HHMM would then be constrained to the estimation of the top-level HMM link structure.

As with the GMM and HMM speech recognition configurations presented earlier in this chapter, the HHMMs could be trained either with speech transcriptions or using only spectral features. The use of higher order, context- and duration-emphasised HMMs for the top-level HMM structure of the HHMM could also be investigated. Figure 5.1 illustrates the relationships between the applicable recognition configurations for SAE classification.

5.2 Speech recognition configurations selected

The primary thesis objective was to develop a SV SAE classification system which operates without speech transcriptions, primarily because the speech transcription process for the AST SAE corpus was not finalised when the thesis implementation stage commenced. The lack of speech transcriptions could be overcome by utilising another English speech corpus such as TIMIT [13]. TIMIT is transcribed at the phoneme level and would provide the necessary data for the development of phoneme recognisers for English speech.

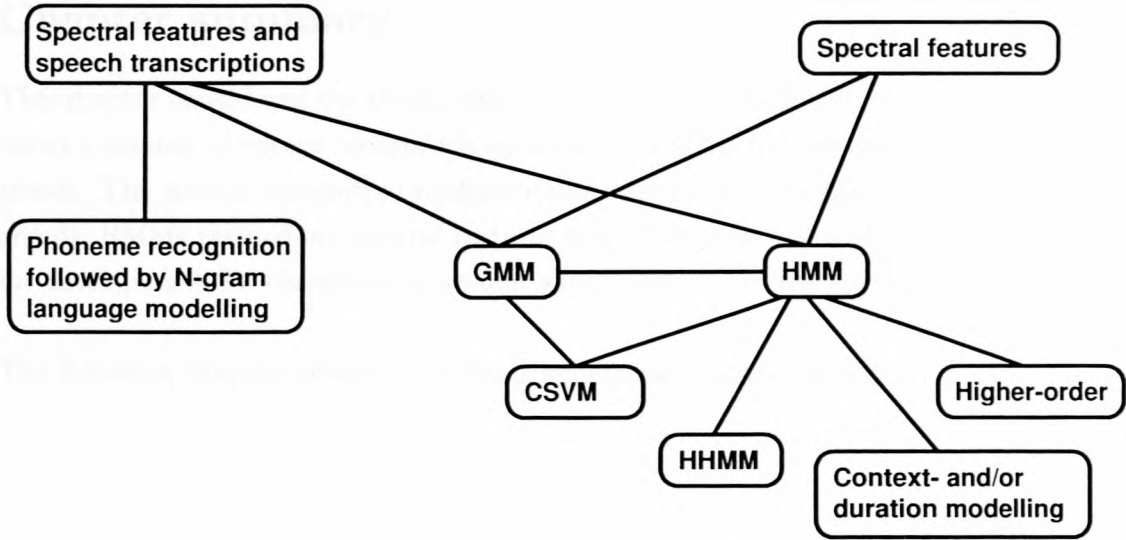


Figure 5.1: *Applicable speech recognition configurations for SAE classification. Shown are the pattern modelling configurations available, possible methods of model initialisation (CSVM, GMM to HMM transform etc.) and whether a pattern modelling configuration is capable of using speech transcriptions. The lines in the figure connect these related concepts.*

Using a non-SAE English corpus in the development of the SAE classification system would bias the speech recognition system in favour of the EE SV, especially if the external speech corpus is representative of American- or British English. This is because the EE SV is considered to be more similar to American- and especially British English than any of the other SAE SVs. The lack of speech transcriptions for the AST SAE corpus and the objections against utilising a non-SAE corpus for speech transcriptions, meant that any speech recognition configuration which requires speech transcriptions was not considered for use in this thesis.

Therefore, it was decided to concentrate on speech recognition configurations which utilise either GMMs or ergodic HMMs trained on the spectral features of the SVs. In order to develop a speech recognition configuration capable of classifying the SAE SVs, each of the components which comprise a typical speech recognition configuration were investigated in turn. These components range from the preemphasis stage (section 4.1) through to the classification stage (section 4.7). Because of time constraints, HHMMs were not investigated.

Chapter summary

This chapter considered the trends which emerged from the literature synopsis and considered a number of speech recognition approaches for SAE SV classification based on these trends. The speech recognition configurations selected for investigation utilise GMMs or ergodic HMMs trained on spectral features only. This is in line with the thesis objectives for developing a SV classification system which does not utilise speech transcriptions.

The following chapter presents the thesis experimental investigation.

Introduction

The experiments presented in this chapter were designed to investigate the performance of different speech recognition systems for classification of speech signals (SV). The experiments were designed to investigate the effect of different speech recognition configurations on the SV classification system. The experiments were designed to investigate the effect of different speech recognition configurations on the SV classification system.

- Speech preprocessing.
- Speech features extraction.
- Feature normalisation.
- SV speech modelling.

A number of different configurations were evaluated for the stages mentioned above. The configurations were evaluated using fixed configurations for the other stages of the system. The results of the experiments were used to evaluate the performance of a number of different SV classification systems. SV speech modelling.

Seven different test segment durations were used in the experiments. The test segments were obtained from the concatenation of multiple utterances. The number of feature frames to concatenate was determined by the frame size of the utterances and the required test segment duration.

Since the average duration of the utterance files examined in the AST-SAE corpus was about 7.6s, test segments were allowed to contain features from utterances of different speakers in order to attain the required test segment duration.

Some test segments contained features from speakers with different genders and some test segments contained features from speakers who utilised different types of telephones. This is because the AST-SAE corpus contains

Chapter 6

Experimental investigation

Introduction

The experiments presented in this chapter were designed to evaluate different speech recognition systems for classification of the SAE SVs. The experiments were designed to investigate one stage of the SV classification system at a time. The stages investigated were:

- Speech preprocessing.
- Speech features extraction.
- Feature normalisation.
- SV speech modelling.

A number of different configurations were utilised for the stage under investigation, while using fixed configurations for the other stages of the system. This resulted in the creation of a number of different SV classification systems for each experiment.

Seven different test segment durations were utilised in the experiments, see Table 6.1. The test segments were obtained from the concatenation of successive feature frames. The number of feature frames to concatenate was determined by the frame skip of 10ms and the required test segment duration.

Since the average duration of the utterance files contained in the AST SAE corpus is about 7,6s, test segments were allowed to contain features from utterance files of different speakers in order to attain the required test segment durations.

Some test segments contained features from speakers with different genders and who utilised different types of telephone. This is because the AST SAE corpus speech was not

Table 6.1: *Test segment durations utilised in experiments.*

Length [s]
2
4
10
30
60
120
300

split according to speaker gender or telephone type, for the reasons given in section 3.2.

The classification experiments were conducted with the separate SV speech models in a parallel bank. Test segments presented to a SV classification system were scored on each of the SV models in turn, with the test segment assigned to the model (SV) producing the highest log-likelihood score.

For each SV model of a classification system, we therefore obtained the number of correct and incorrect classifications on a given test segment duration. From the number of correct and incorrect classifications the Recognition Error Rate (RER) was computed for each of the SV models of a SV classification system. For a SV classification system, an average RER value was obtained for each test segment duration by taking the average of the RER values attained by the system's SV models on the given test segment duration.

By taking the average of the average RER values, the so-called 'overall average RER' was obtained for each system. The overall average RER was used to determine the effect on the SV classification system's performance as different configurations were utilised for a specific stage of the speech recognition system. In this manner a single figure of merit was obtained in terms of which different SV classification systems could be compared.

Each of the SV classification systems investigated were given a unique label which reflects the speech recognition stage investigated and the specific configuration used by the particular classification system.

The difference between the SV classification systems will be presented graphically, in terms of the overall average RER figures attained by each system. Where applicable, the detailed experimental results will be presented in section B.2, and references will be

provided to these results after the presentation of the overall average RER comparison graph and table.

The detailed results will be presented in both tabular and graphical format. The detailed classification results will include the RER values attained by the individual SV models of a classification system as well as the average RER values of the system. The tables and graphs in section B.2 will indicate the lower and upper limit of the average RER values for a 95% confidence interval. The confidence intervals for the average RER values were computed with the assumption that the average RER values were obtained from independent Bernoulli trials. The procedure for computing the confidence intervals is briefly described in section B.1.

For each experiment conducted, the following will be indicated:

- Speech *preprocessing* configuration.
- Speech *features* used.
- Feature *normalisation* configuration.
- Speech *modelling* used.
- *Results* obtained.
- *Interpretation* of experimental results.

The experiments are presented in the following sections, where the sections are devoted to the specific stages of the SV classification process investigated:

- Speech preprocessing experiment (section 6.1).
- Speech features experiment (section 6.2).
- Feature normalisation experiment (section 6.3).
- Speech modelling experiments (section 6.4).

Some of the experimental strategies were revised or devised as new experimental results were gathered. If a given configuration of a processing stage of the system was found to improve the system overall average RER, this configuration would be incorporated in subsequent experiments. This incorporation was performed in as logical a manner as possible.

For instance, during the speech features extraction experiment it was determined that

using PLP cepstral coefficients instead of the MFCCs used in the preprocessor experiment, the overall average RER was reduced by 11,9%. Only the speech features extraction configuration was changed, while the rest of the configurations were retained unchanged from the ‘best’ classification system of the preprocessor experiment.

6.1 Speech preprocessing experiment

The speech preprocessing experiment investigated two different speech preprocessing configurations in terms of the overall average RER attained with each configuration. The preprocessing configuration which yielded the lower overall average RER would be used as preprocessor configuration in subsequent experiments.

In accordance with the experimental procedure described in the beginning of this chapter, only the preprocessing stage configurations were varied in this experiment. The rest of the system stages; namely the speech features extraction-, feature normalisation- and SV speech modelling stages were kept to fixed configurations for the duration of the experiment.

Preprocessing

Preprocessing consisted of one of the following configurations:

- A CARP with the following parameters:

- $win_{dur} = 0,2s$.
- $ret_{dur} = 0s$.

The CARP was followed by preemphasis and power normalisation. This preprocessor configuration was labelled CARP.

- A PFP prepended to the CARP configuration. The PFP had the following parameters:

- The second percentile was used for p_1 and p_2 .
- $ret_{dur} = 0,5s$.
- $peak_{min} = 0,06s$.
- $gap_{min} = 0,3s$.
- $frm_{dur} = 0,02s$.
- $frm_{ovl} = 0,01s$.
- $o_1 = 3$.

$$- o_2 = 0,25.$$

This preprocessor configuration was labelled PFP.

Features

The speech features utilised for this experiment consisted of 18-dimensional MFCCs. The MFCCs were computed using twenty-two filter bands, with the analysis frequency range varying between 200Hz and 3.5kHz.

Normalisation

Normalisation consisted of feature scaling.

Modelling

SV modelling consisted of one eighteen-dimensional (to match the feature vector dimension) diagonal covariance Gaussian PDF per SV. The model parameters were estimated via MLE.

Results

The classification results attained with the different preprocessor configurations are shown in Figure 6.1. The overall average RER attained by each of the configurations is shown in Table 6.2. The detail classification results are presented in Table B.1, Figure B.1, Table B.2 and Figure B.2.

Interpretation

The PFP preprocessor configuration reduced the overall average RER attained with the CARP configuration by 7,78%, from 56,44% to 52,05%. The difference between these preprocessor configurations lies in the addition of the PFP. The improvement in overall average RER with the addition of the PFP corresponds with the observations of Rabiner and Juang [20] that non-speech events (and the mis-detection thereof) may detract from the overall system classification performance. The PFP preprocessor configuration was

Table 6.2: Overall average RER [%] for the preprocessor configurations investigated.

Preprocessor configuration	Overall average RER [%]
CARP	56,44
PFP	52,05

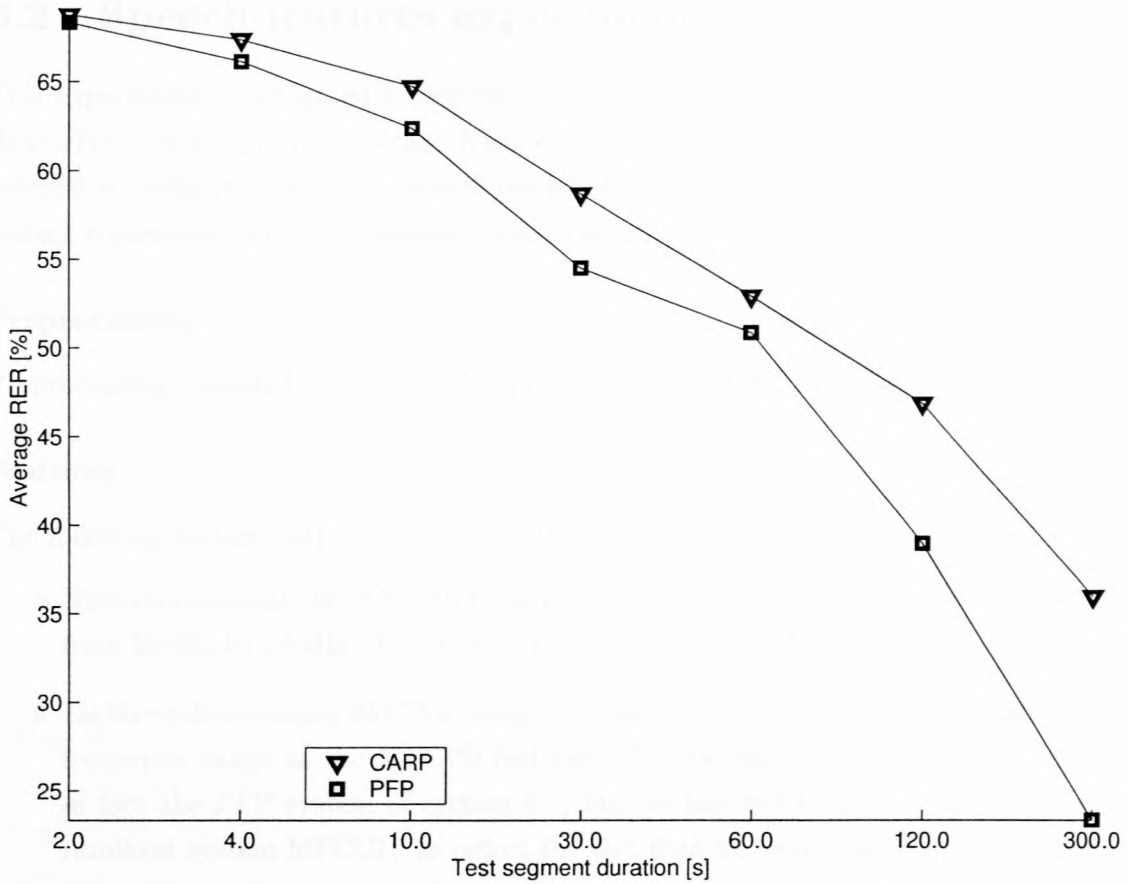


Figure 6.1: Comparison of the average RER [%] attained with two different preprocessor configurations. The CARP configuration utilised a CARP preprocessor followed by preemphasis and power normalisation. The PFP configuration prepended a PFP preprocessor to the CARP configuration.

used throughout the remainder of the thesis experiments.

It will be observed from the detail classification results shown in Table B.1, Figure B.1, Table B.2 and Figure B.2; that the overall average RER figures for the different SVs differed considerably. For both the CARP and PFP configurations, the initial best (lowest) and worst (highest) RER figures for the SVs differ by about 50%. It is postulated that this is caused by speaker gender- and telephone channel type mismatches between the SVs.

As mentioned in section 3.2; we would have utilised gender- and telephone channel-specific SV models had the speaker gender- and telephone channel type been known for all the utterances in the SAE corpus. It is postulated that use of these gender- and telephone channel-specific models would reduce the difference in the RER figures of the different SVs and therefore improve the classification performance of the system.

6.2 Speech features experiment

This experiment investigated a number of speech feature extraction configurations and their effect on the overall average RER of the SV classification system. The feature extraction configuration which yielded the lowest overall average RER was chosen as the feature representation for subsequent experiments.

Preprocessing

Preprocessing consisted of the the PFP preprocessor configuration (section 6.1).

Features

The following feature extraction configurations were investigated:

- Nine-dimensional MFCCs with twenty-two filter bands and analysis frequency range from 200Hz to 3.5kHz. This configuration was labelled MFCC9.
- Eighteen-dimensional MFCCs using the same number of filter bands and analysis frequency range as the MFCC9 features. The resultant classification system was in fact the PFP system of section 6.1, but we labelled this configuration and the resultant system MFCC18 to reflect the fact that we were investigating the feature extraction configuration in this experiment.
- Formant frequencies and bandwidths (F1, F2, F3, B1, B2, B3) computed from eighth-order Linear Prediction (LP) filter polynomial, yielding six-dimensional feature vectors. This configuration was labelled FF.
- A feature set consisting of the combination of the MFCC9 and FF feature sets, resulting in a feature vector dimension of fifteen. This configuration was labelled MFCC9_FF.
- Features computed from the filter bank spectral analysis setup proposed by Arslan and Hansen. This configuration was labelled AH. These features were of dimension sixteen.
- Eighth-order PLP cepstral coefficients with frame energy, labelled PLP. The resultant feature vector dimension was nine.

An eighth-order LP filter polynomial (eight filter poles) was utilised in the FF, MFCC9_FF and PLP configurations. This corresponds partially to the guideline for LP filter analysis given by [18]. This rule of thumb recommends the use of one pole per kHz of sampling frequency plus two to four additional poles to cater for various effects. For this thesis, one pole per kHz of sampling frequency (8kHz) was deemed to be sufficient.

Normalisation

Normalisation consisted of feature scaling.

Modelling

Each SV was modelled by one diagonal covariance Gaussian. The model parameters were estimated via MLE. The model dimension depended only on the feature vector dimension of the feature extraction configuration used, since feature scaling normalisation does not alter the dimension of the input feature vector.

Results

Figure 6.2 shows the classification results obtained with the different feature extraction configurations. Table 6.3 shows the overall average RER attained with the different configurations. Detail classification results are presented in Table B.3-Table B.8 and Figure B.3-Figure B.8.

Table 6.3: *Overall average SV RER [%] for the speech feature extraction configurations investigated. MFCC9 and MFCC18 were nine- and eighteen-dimensional MFCCs respectively, computed from twenty-two filter bands over an analysis frequency range of 200Hz to 3.5kHz. FF consisted of formant frequencies computed from an eighth-order LP filter polynomial. MFCC9_FF consisted of the combination of the MFCC9 and FF feature extraction configurations. The AH configuration utilised the filter bank spectral analysis method of Arslan and Hansen. The PLP configuration consisted of eight-dimensional PLP cepstral coefficients with frame energy as additional feature.*

Feature extraction configuration	Overall average RER [%]
MFCC9	51,51
MFCC18	52,05
FF	56,18
MFCC9_FF	50,64
AH	53,18
PLP	45,86

Interpretation

From Figure 6.2 and Figure 6.2 it is seen that the PLP feature extraction configuration yielded the best classification performance. The PLP system reduced the overall average RER of the PFP system (section 6.1) by 11,9%, from 52,05% to 45,86%. The only difference between the two systems was the use of PLPs in the PLP system instead of the

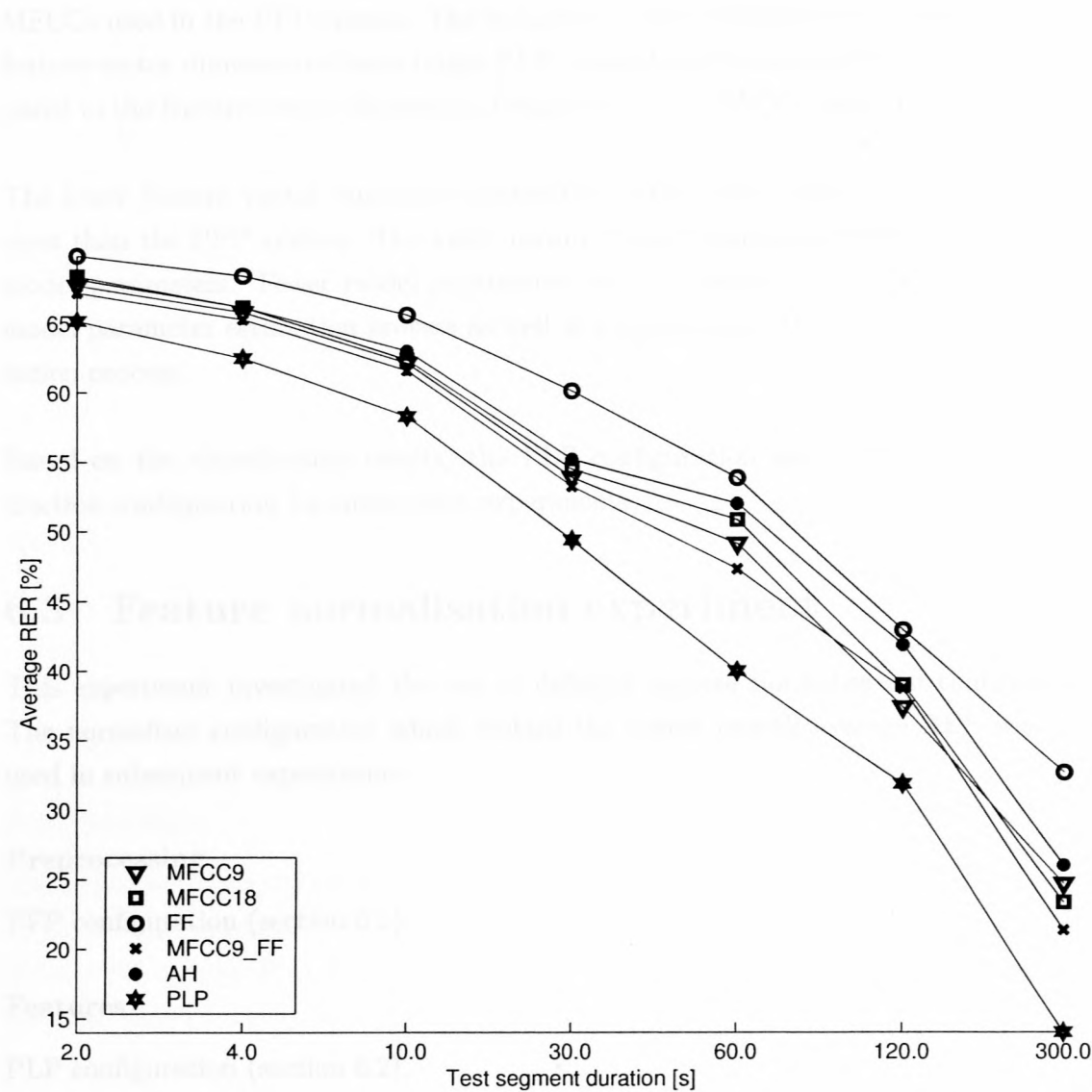


Figure 6.2: Average RER [%] for the speech feature extraction configurations investigated. MFCC9 and MFCC18 were nine- and eighteen-dimensional MFCCs respectively, computed from twenty-two filter bands over an analysis frequency range of 200Hz to 3.5kHz. FF consisted of formant frequencies computed from an eighth-order LP filter polynomial. MFCC9_FF consisted of the combination of the MFCC9 and FF feature extraction configurations. The AH configuration utilised the filter bank spectral analysis method of Arslan and Hansen. The PLP configuration consisted of eight-dimensional PLP cepstral coefficients with frame energy as additional feature.

MFCCs used in the PFP system. The reduction in overall average RER was attained at a feature vector dimension of nine (eight PLP cepstral coefficients and frame energy), compared to the feature vector dimension of eighteen for the MFCCs used in the PFP system.

The lower feature vector dimension makes the PLP system computationally more efficient than the PFP system. The lower feature vector dimension reduced the number of model parameters. Fewer model parameters implies a reduction in the duration of the model parameter estimation process as well as a reduction in the duration of the classification process.

Based on the classification results, the PLP configuration was selected as feature extraction configuration for subsequent experiments.

6.3 Feature normalisation experiment

This experiment investigated the use of different feature normalisation configurations. The normaliser configuration which yielded the lowest overall average RER was to be used in subsequent experiments.

Preprocessing

PFP configuration (section 6.1).

Features

PLP configuration (section 6.2).

Normalisation

The normaliser configurations, with their labels in brackets, were:

- No normalisation (NONE).
- Feature scaling (FS). The resultant classification system was in fact the PLP system used in the feature extraction experiment (section 6.2). We assigned a new label to this configuration (and the resultant classification system) to indicate that the normaliser configuration was investigated in this experiment.
- KLT with no dimension reduction (KLT).
- Δ coefficients (Δ _KLT) followed by KLT with dimension retention factor of 0,95.
- $\Delta\Delta$ coefficients ($\Delta\Delta$ _KLT) followed by KLT with dimension retention factor of 0,95.

- FFE with three vectors per frame, one vector spacing, followed by KLT with dimension retention factor of 0,95 (FFE_KLT).
- FCE of order two, followed by KLT with dimension retention factor of 0,95 (FCE_KLT).

Modelling

Speech modelling consisted of one diagonal covariance Gaussian PDF per SV. The model parameters were estimated via MLE. The model feature vector dimension was determined by the output feature vector of the specific feature normalisation configuration under consideration.

Results

Figure 6.3 shows the classification results obtained with the different feature normaliser configurations. The overall average RER attained by each of the normaliser configurations is shown in Table 6.4. Detail classification results are presented in Table B.9-Table B.15 and Figure B.9-Figure B.15.

Interpretation

Use of the KLT normaliser in the KLT configuration, instead of the feature scaling used in the FS configuration (PFP system), resulted in a reduction of 6,39% in the overall average RER; from 45,86% to 42,93%. A number of observations may be made regarding the outcome of this experiment.

Using feature scaling (FS configuration) resulted in the same classification results as using no normalisation (configuration NONE). This is because feature scaling does not alter the information content of the features. It reduces the sensitivity of the features to numerical instabilities during subsequent processing.

The classification results obtained using delta coefficients (Δ _KLT and $\Delta\Delta$ _KLT configurations) were somewhat worse than expected. Perhaps the temporal behaviour of the features (as measured by the delta coefficients), was not distinguishable with the non-temporal modelling applied in this and previous experiments. The Δ _KLT and $\Delta\Delta$ _KLT normalisers were applied to the features of individual utterances, and not the concatenated feature blocks of the test segments used in classification.

The results obtained with the KLT configuration indicate that the KLT succeeded in decorrelating the feature vector dimensions to a significant degree. This decorrelation

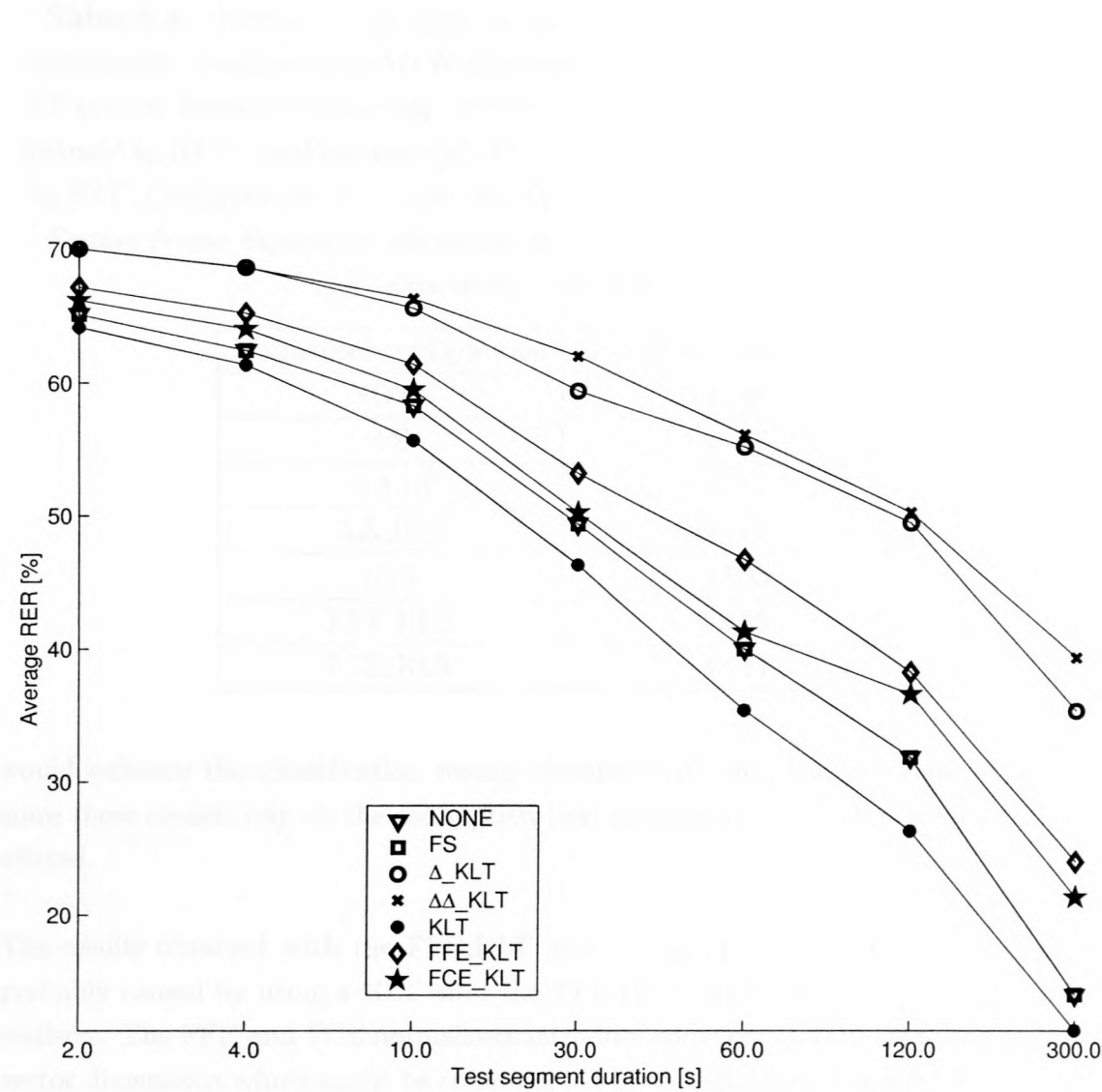


Figure 6.3: Average RER [%] for the feature normalisation configurations investigated. Configuration *NONE* performed no feature normalisation. Configuration *FS* utilised Feature Scaling only. Configuration Δ _KLT used Δ (velocity) coefficients followed by KLT. Configuration $\Delta\Delta$ _KLT used $\Delta\Delta$ (acceleration) coefficients followed by KLT. Configuration *KLT* used only the KLT as normalisation. *FFE_KLT* utilised Feature Frame Expansion followed by KLT. *FCE_KLT* utilised Feature Cross-term Expansion followed by KLT.

Table 6.4: Overall average RER [%] for the feature normalisation configurations investigated. Configuration *NONE* performed no feature normalisation. Configuration *FS* utilised Feature Scaling only. Configuration Δ -KLT used Δ (velocity) coefficients followed by KLT. Configuration $\Delta\Delta$ -KLT used $\Delta\Delta$ (acceleration) coefficients followed by KLT. Configuration *KLT* used only the KLT as normalisation. *FFE_KLT* utilised Feature Frame Expansion followed by KLT. *FCE_KLT* utilised Feature Cross-term Expansion followed by KLT.

Normaliser configuration	Overall average RER [%]
NONE	45,86
FS	45,86
Δ -KLT	57,67
$\Delta\Delta$ -KLT	58,952
KLT	42,93
FFE_KLT	50,85
FCE_KLT	48,44

would enhance the classification results obtained with diagonal covariance Gaussians, since these models rely on the assumption that the feature vector dimensions are uncorrelated.

The results obtained with the FFE_KLT and FCE_KLT normaliser configurations are probably caused by using a KLT after the FFE (FFE_KLT) and FCE (FCE_KLT) normalisers. The FFE and FCE normalisers introduce some correlation between the feature vector dimensions which might be counteracted by the application of a KLT.

6.4 SV speech modelling experiments

The experiments presented in this section investigated different SV speech modelling configurations. Gaussian PDFs were used in the preceding experiments because of their simplicity, so that the SV speech modelling introduced as little complexity in the SV classification system as possible. This presented a common speech modelling basis for the comparison of different speech preprocessing-, feature extraction- and feature normalisation configurations.

In this section of the experimental investigation chapter, more complex speech modelling configurations were investigated. The experiments conducted in this section focused on speech modelling configurations which were based on GMMs and ergodic HMMs.

The GMM experiments consisted of the following:

- An experiment which investigated the use of GMMs to model the SVs and the influence of GMM cluster count on the overall average RER of the configuration (subsection 6.4.1).
- Investigating the use of a CSVM (Common SV Model) GMM, trained on the features of all variants, as initialisation model for the SV GMMs (subsection 6.4.2).

The HMM experiments were:

- Investigating first-order HMM configurations for SV modelling (subsection 6.4.3). These configurations utilised different methods to initialise the HMM parameters.
- Using GMMs as emitting densities in the SV HMMs (subsection 6.4.4).
- Investigating higher-order HMM configurations for SV modelling (subsection 6.4.5). The higher-order HMM configurations differed in terms of the type of higher-order HMM representation used as well as the initialisation used for the HMM parameters.
- Investigating a third-order HMM configuration for SV modelling (subsection 6.4.6).
- Using full covariance Gaussian PDFs as SV HMM emitting densities (subsection 6.4.7).

Throughout the HMM experiments ergodic (fully-connected) HMMs were used, since the temporal nature of the SVs was unknown.

6.4.1 Modelling SVs with GMMs

This experiment determined whether the system overall average RER could be lowered if GMMs were used to model the SVs instead of the single Gaussian PDFs used in previous experiments. GMMs were investigated for modelling of the SVs, since the SV feature space may be too complex for modelling SVs with a single Gaussian PDF. The role of the GMM cluster count on the system overall average RER was also investigated.

Preprocessing

PFP configuration (section 6.1).

Features

PLP configuration (section 6.2).

Normalisation

KLT configuration (section 6.3).

Modelling

Speech modelling for this experiment consisted of a single GMM per SAE SV. The GMM component counts were varied between two and hundred-and-twenty-eight components in steps of two. Each GMM component PDF consisted of a single diagonal covariance Gaussian PDF of dimension nine (the feature vector dimension following normalisation). The GMM component centroids were initialised via VQ codebook. A separate VQ codebook was used for each SV. Training of the models was accomplished via five iterations of the EM-algorithm.

Results

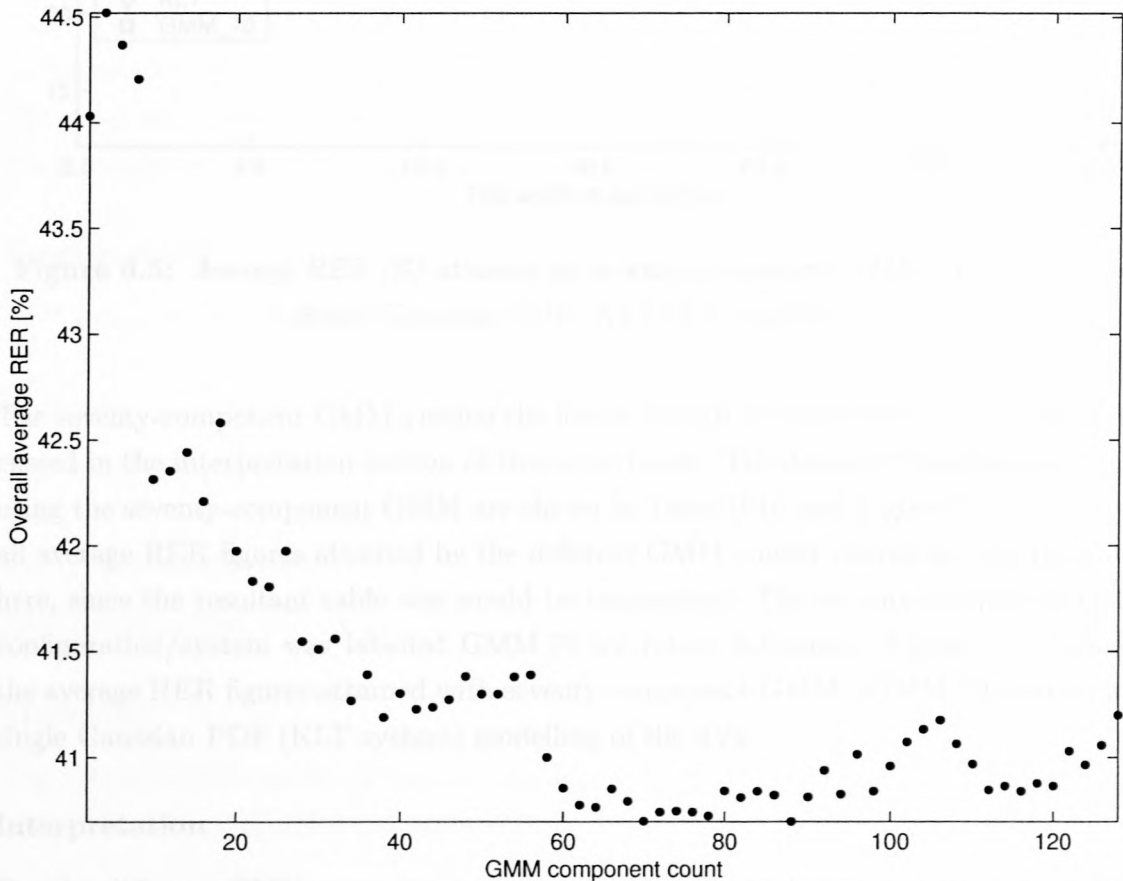


Figure 6.4: Overall average RER [%] as a function of varying the component count for diagonal covariance GMMs.

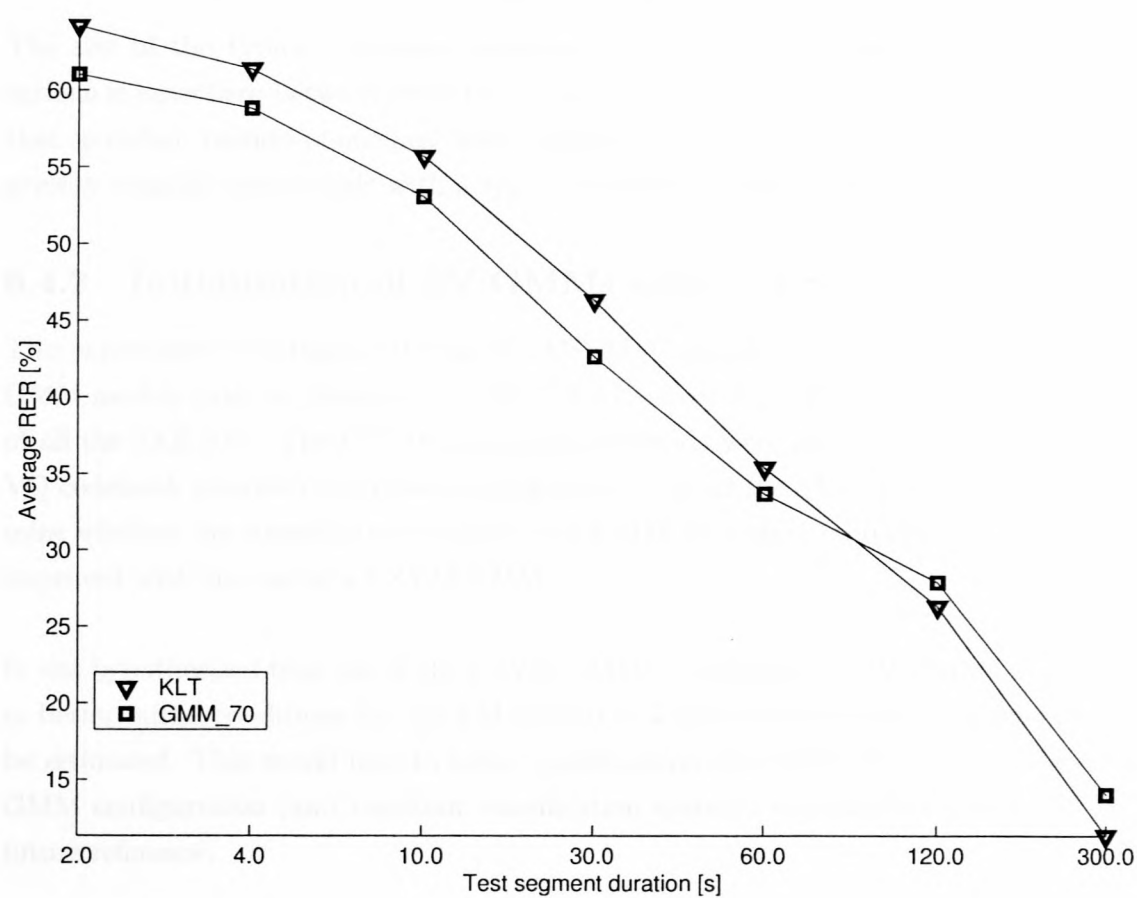


Figure 6.5: Average RER [%] attained by seventy-component GMM- (GMM_70) and single Gaussian PDF (KLT) SV modelling.

The seventy-component GMM yielded the lowest overall average RER. This result is discussed in the interpretation section of this experiment. The detailed classification results using the seventy-component GMM are shown in Table B.16 and Figure B.16. The overall average RER figures attained by the different GMM cluster counts are not tabulated here, since the resultant table size would be impractical. The seventy-component GMM configuration/system was labelled GMM_70 for future reference. Figure 6.5 compares the average RER figures attained with seventy-component GMM- (GMM_70 system) and single Gaussian PDF (KLT system) modelling of the SVs.

Interpretation

For the different GMM component counts considered, the lowest overall average RER was attained using a component count of seventy. Using seventy-component GMMs to model the SVs resulted in a RER reduction of 3,14%, from the 42,93% attained by the SV classification system which utilised single Gaussian PDFs (KLT), to 41,58%.

The size of the typical phoneme inventory used for English speech recognition applications is anywhere between forty to sixty phonemes. If the assumption was maintained that so-called ‘pseudo-phonemes’ were modelled here, then a GMM component count of seventy roughly corresponds with a typical phoneme inventory size.

6.4.2 Initialisation of SV GMMs using a CSVM GMM

This experiment investigated the use of a CSVM (Common SV Model) to initialise the SV GMM models prior to training. The CSVM consisted of a GMM trained on the features of all the SAE SVs. The CSVM component centroids were initialised via a seventy-level VQ codebook obtained from the training features of all the SVs. The idea was to determine whether the classification results of the GMM_70 system (subsection 6.4.1) could be improved with the use of a CSVM GMM.

It was hypothesised that use of the CSVM GMM to initialise the SV GMMs would result in better initial conditions for the EM-algorithm when the SV GMM parameters were to be estimated. This would lead to better classification results for the system. The CSVM GMM configuration (and resultant classification system) was labelled GMM_CSVM for future reference.

Preprocessing

PFP configuration (section 6.1).

Features

PLP configuration (section 6.2).

Feature normalisation

KLT configuration (section 6.3).

Modelling

The GMM_CSVM configuration differed from the GMM_70 configuration only in terms of the CSVM initialisation it utilised.

The CSVM GMM was trained on the features of all the SVs, using five iterations of the EM-algorithm. Each of the SV GMMs started out as exact copies of the trained CSVM GMM. The SV GMMs were then retrained using the training features of each specific SV and five iterations of the EM-algorithm.

Results

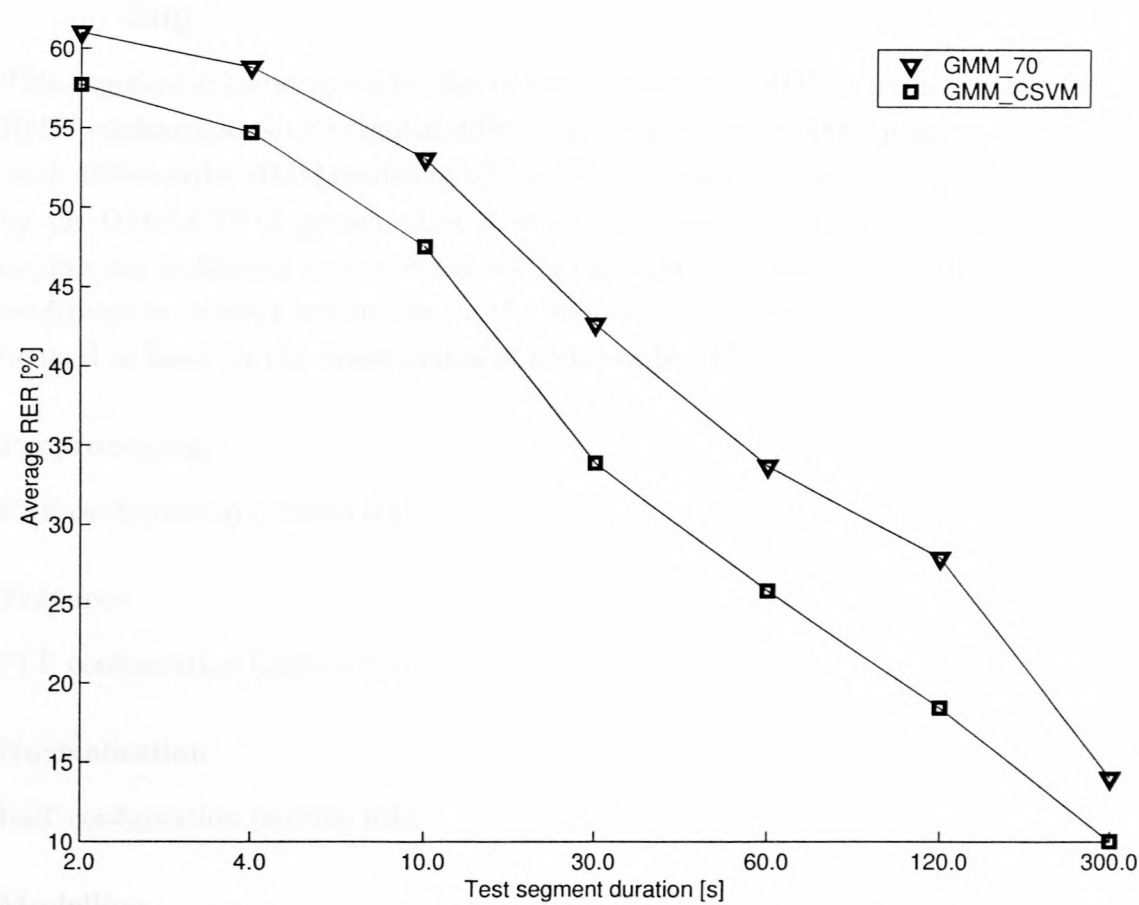


Figure 6.6: Average RER [%] for ‘standard’ 70-component SV GMMs (GMM_70) vs. that attained by 70-component CSVM-initialised SV GMMs (GMM_CSVM).

Figure 6.6 compares the overall average RER attained by ‘standard’ seventy-component GMMs (GMM_70 configuration) and seventy-component CSVM-initialised SV GMMs (GMM_CSVM configuration). The detail classification results obtained by the GMM_CSVM configuration are shown in Table B.17 and Figure B.17.

Interpretation

Use of a CSVM GMM to initialise the SV GMMs (GMM_CSVM configuration) reduced the overall average RER attained by the ‘standard’ GMM models (GMM_70 configuration) by 14,77%, from 41,58% to 35,44%. This experimental result highlighted the importance of adequate SV model initialisation prior to parameter estimation using the EM-algorithm.

6.4.3 Investigating first-order HMM configurations for SV modelling

This experiment investigated the use of first-order ergodic HMMs to model the SVs. The HMM configurations investigated differed in terms of the HMM parameter initialisation used. If first-order HMM modelling of the SVs improved the overall average RER attained by the GMM_CSVM system, then it would have indicated that modelling of the SVs require the additional temporal modelling capabilities provided by HMMs. The HMM configuration (if any) responsible for the improvement in the overall average RER would be used as basis for the investigation of higher-order HMM configurations.

Preprocessing

PFP configuration (section 6.1).

Features

PLP configuration (section 6.2).

Normalisation

KLT configuration (section 6.3).

Modelling

The number of states used in the first-order HMMs was set equal to the number of component densities used in the GMM experiments, in order to compare GMM and HMM configurations with each other. All HMMs in this experiment used this number of states (seventy). The first-order HMM configurations investigated (with their labels in brackets) were:

- Conventional ergodic HMMs (X1).
- HMMs initialised from CSVM HMM (X1_CSVM).
- HMMs initialised from SV GMMs (X1_GMM).
- HMMs initialised from a CSVM HMM; which in turn was initialised from the CSVM GMM of the GMM_CSVM configuration (X1_GMM_CSVM).

The X1 configuration used conventional first-order ergodic HMMs with seventy states. The HMMs contained a single diagonal covariance Gaussian PDF per HMM state; with the PDF centroids initialised via VQ codebook. Separate codebooks were used for each SV. The PDF variances were initialised to small initial values. The HMM link weights

were initialised to values proportional to the number of states in the HMM. No class-specific prior information was built into the HMM link structures.

The links emanating from the entry states of the HMMs were weighted equally. These link weights were fixed for the lifetime of the HMM, i.e. training of the HMMs did not alter these weights. The link weights of the internal HMM states were initialised to equal weights, except for the self-loop weight of each state and the weights of the links leading to the HMM exit state.

Each self-loop weight was initialised to a larger value than the rest of the state link weights, with this value a function of the number of HMM states. The weights of links leading to the exit state of the HMM were fixed for the lifetime of the HMM, in order to ensure a valid path through the HMM during the parameter estimation process.

The X1_CSVM configuration utilised a CSVM HMM to initialise each of the SV HMMs. The CSVM HMM was initialised in the same manner as the HMMs of configuration X1, but a single VQ codebook was utilised to initialise the CSVM HMM PDF centroids. This single VQ codebook was obtained from the training features of all the SVs. The SV HMMs were initialised as copies of the trained CSVM HMM and then retrained using only the training features of the corresponding SV.

Configuration X1_GMM utilised SV HMMs which were initialised from the trained GMMs of the GMM_70 configuration. A GMM may be represented by an ergodic HMM equivalent. This was discussed in subsection 4.5.3.

Configuration X1_GMM_CSVM utilised the GMM to HMM transformation employed in configuration X1_GMM to convert the GMM CSVM of configuration GMM_CSVM to an equivalent HMM CSVM model. This CSVM HMM was then retrained on the features of all SVs, after which the same process employed in configuration X1_CSVM was used to initialise and train the SV models.

Results

Figure 6.7 shows the average RER values attained by each of the first-order HMM configurations investigated. Table 6.5 shows the overall average RER for the first-order HMM configurations investigated. The detail classification results for the first-order HMM configurations investigated appear in Table B.18-Table B.21 and Figure B.18-Figure B.21.

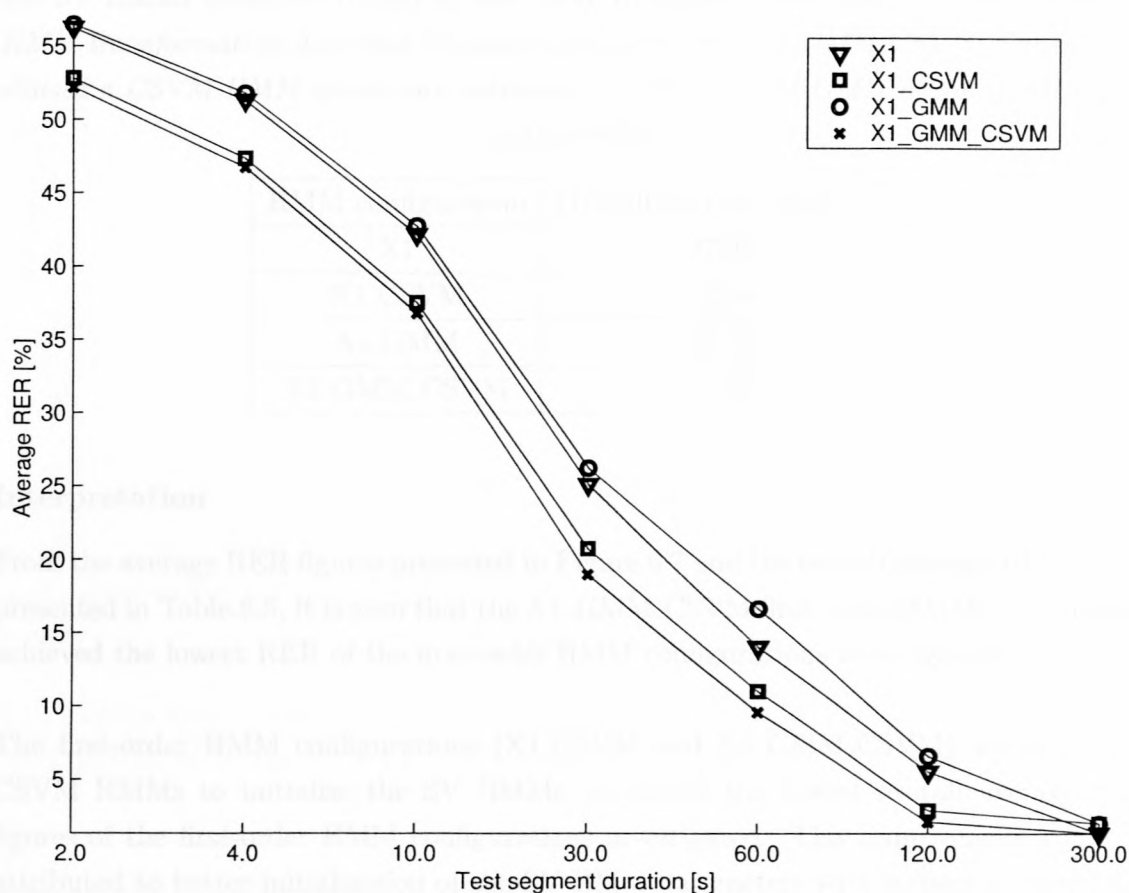


Figure 6.7: Average RER [%] for the first-order HMM configurations investigated.

Configuration *X1* utilised conventional seventy-state ergodic HMMs with a single diagonal covariance Gaussian PDF per HMM state. Configuration *X1_CSVM* utilised a CSVM HMM to initialise the SV HMMs. Configuration *X1_GMM* initialised the SV HMMs using the GMMs of the GMM_70 configuration and the GMM to ergodic HMM transformation described in subsection 4.5.3. The *X1_GMM_CSVM* configuration utilised a CSVM HMM which was initialised from the CSVM GMM of the GMM_CSVM configuration.

Table 6.5: Overall average RER [%] for the first-order HMM configurations investigated. Configuration X1 utilised conventional seventy-state ergodic HMMs with a single diagonal covariance Gaussian PDF per HMM state. Configuration X1_CSVM utilised a CSVM HMM to initialise the SV HMMs. Configuration X1_GMM initialised the SV HMMs using the GMMs of the GMM_70 configuration and the GMM to ergodic HMM transformation described in subsection 4.5.3. The X1_GMM_CSVM configuration utilised a CSVM HMM which was initialised from the CSVM GMM of the GMM_CSVM configuration.

HMM configuration	Overall average RER [%]
X1	27,95
X1_CSVM	24,90
X1_GMM	28,92
X1_GMM_CSVM	23,95

Interpretation

From the average RER figures presented in Figure 6.7 and the overall average RER figures presented in Table 6.5, it is seen that the X1_GMM_CSVM first-order HMM configuration achieved the lowest RER of the first-order HMM configurations investigated.

The first-order HMM configurations (X1_CSVM and X1_GMM_CSVM) which utilised CSVM HMMs to initialise the SV HMMs, produced the lowest overall average RER figures of the first-order HMM configurations investigated. This improvement could be attributed to better initialisation of the SV HMM parameters with respect to the global (corpus-wide) temporal and spectral characteristics.

The X1_GMM_CSVM configuration faring better than the X1_CSVM configuration was probably the result of the X1_GMM_CSVM configuration CSVM HMM spectral modelling component being initialised from the trained CSVM GMM of the GMM_CSVM system.

Figure 6.8 compares the average RER attained by GMM modelling (GMM_CSVM configuration) and HMM modelling (X1_GMM_CSVM configuration) of the SVs. The X1_GMM_CSVM configuration lowered the RER attained by the GMM_CSVM configuration by 32,4%, from 35,44% to 23,95%.

The X1_GMM_CSVM configuration was used as the basis for subsequent experiments involving higher-order HMM configurations.

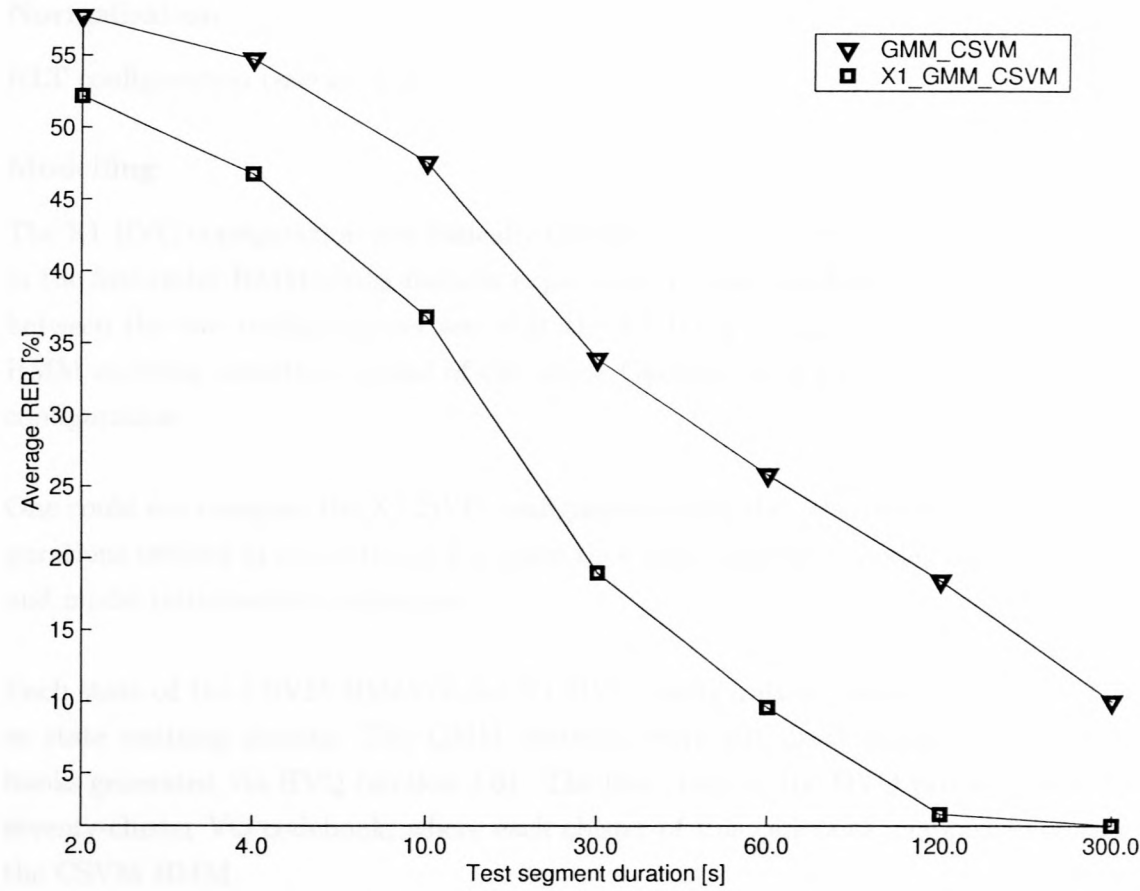


Figure 6.8: Comparison of the average RER [%] attained by GMM modelling (GMM_CSVM configuration) and HMM modelling (X1_GMM_CSVM configuration) of the SVs.

6.4.4 Using GMMs as emitting densities in the SV HMMs

This experiment investigated the use of GMMs as emitting densities in the SV HMMs. It was postulated that the single Gaussian PDFs used in the HMMs of the previous experiment (subsection 6.4.3) provide inadequate modelling of the SV spectral characteristics. For this reason, the use of GMMs as HMM emitting densities was considered. The configuration thus created was labelled X1_HVQ.

Preprocessing

PFP configuration (section 6.1).

Features

PLP configuration (section 6.2).

Normalisation

KLT configuration (section 6.3).

Modelling

The X1_HVQ configuration was basically the same as the X1_CSVM configuration utilised in the first-order HMM configurations experiment (subsection 6.4.3). The only difference between the two configurations was that the X1_HVQ configuration utilised GMMs as HMM emitting densities instead of the single Gaussian PDFs utilised in the X1_CSVM configuration.

One could not compare the X1_HVQ configuration with the other first-order HMM configurations utilised in subsection 6.4.3, since they utilise different modelling configurations and model initialisation techniques.

Each state of the CSVM HMM of the X1_HVQ configuration contained a unique GMM as state emitting density. The GMM centroids were initialised using unique VQ codebooks generated via HVQ (section 4.6). The first stage of the HVQ process generated a seventy-cluster VQ codebook, where each cluster of this codebook mapped to a state of the CSVM HMM.

The first-stage HVQ codebook was generated from the pooled features of all the SVs. The second stage of the HVQ process clustered the initial seventy clusters into four clusters each, producing seventy unique four-cluster VQ codebooks. These four-cluster codebooks were used to initialise the GMM state emitting densities of the X1_HVQ configuration CSVM HMM. The GMM component count of four utilised in this experiment was chosen arbitrarily.

Model parameters were estimated using five iterations of the Viterbi algorithm.

Results

The detailed classification results attained with the X1_HVQ configuration are shown in Table B.22 and Figure B.22.

Interpretation

The overall average RER attained by the X1_HVQ configuration was 24,27%. This was higher than the overall average RER (23,95%) attained by the best first-order HMM

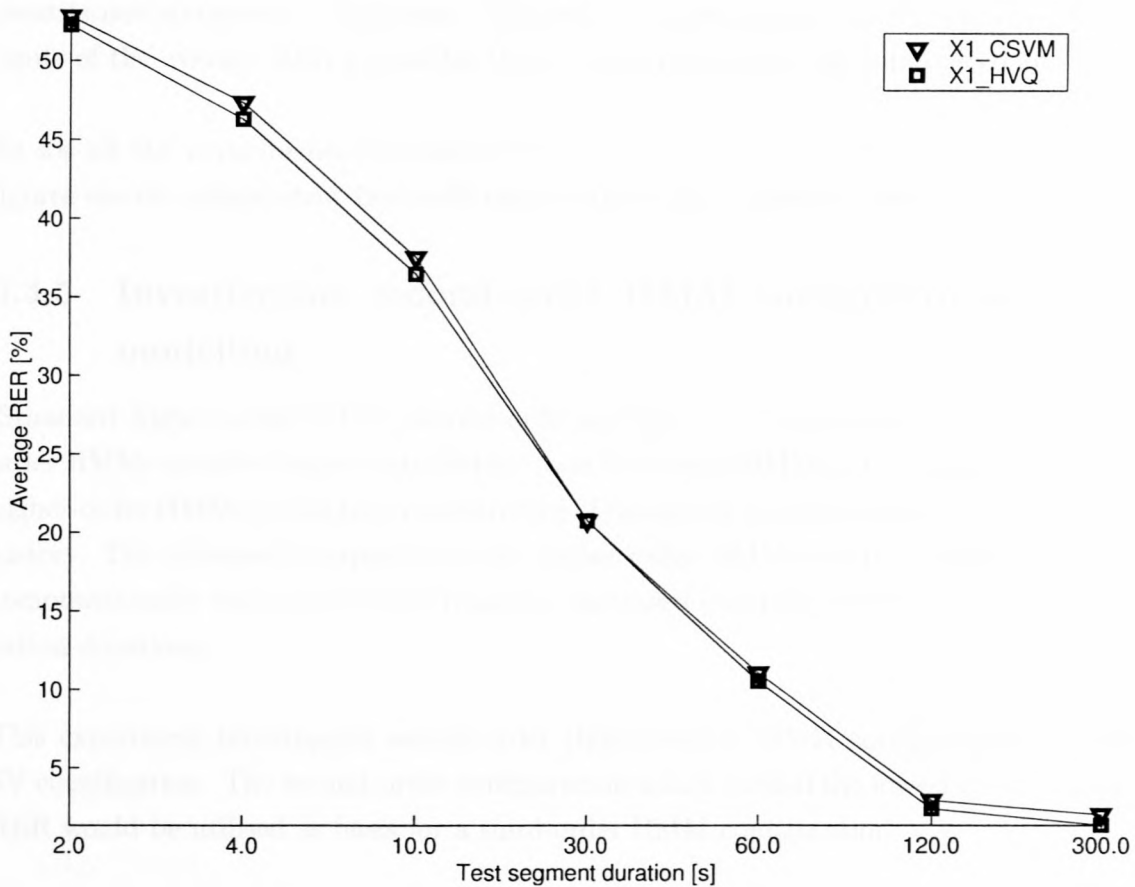


Figure 6.9: Average RER [%] attained by HMM configuration which utilised GMMs as state emitting densities (X1_HVQ configuration) vs. that of the HMM configuration which utilised single Gaussian PDFs as state emitting densities (X1_CSVM configuration).

configuration, X1_GMM_CSVM, but these two configurations are too dissimilar for comparison.

As mentioned before, the X1_CSVM- and X1_HVQ configurations differ only in terms of the type of emitting densities used in their HMMs (single Gaussian PDF vs. four-component GMM). Therefore, by comparing the overall average RER attained by these two configurations, it was possible to determine the effect of the HMM emitting density type on the configuration overall average RER. See Figure 6.9 for a comparison of the average RER attained by the two configurations.

The X1_HVQ configuration overall average RER was a 2,53% improvement on the 24,9% overall average RER of the X1_CSVM configuration. A comparison of the average RER figures for these two configurations (Table B.19 vs Table B.22) indicates that the improve-

ment is not statistically significant. This will be more apparent if the upper and lower limits of the average RER figures for these configurations are taken into account.

As for all the experiments conducted, the upper and lower limits of the average RER figures are for independent Bernoulli trials with a 95% confidence level.

6.4.5 Investigating second-order HMM configurations for SV modelling

‘Standard’ higher-order HMMs (second-order and above) and duration-emphasised higher-order HMMs model a longer state history than first-order HMMs do. Context-emphasised higher-order HMMs model the co-occurrence of categories in addition to an increased state history. The increased complexity of the higher order HMMs result in them being more computationally expensive, which results in increased parameter estimation- and classification durations.

This experiment investigated second-order (higher-order) HMM configurations for SAE SV classification. The second-order configuration which yielded the lowest overall average RER would be utilised as basis for a third-order HMM configuration.

Theoretically, one should be able to increase the HMM order until either the RER starts to increase, indicating a data scarcity problem, or until the model size is too large for the computer hardware (memory and CPU combined) to handle.

The second-order HMM configurations investigated here used the X1_GMM_CSVM first-order HMM configuration as basis.

Preprocessing

PPF configuration (section 6.1).

Features

PLP configuration (section 6.2).

Normalisation

KLT configuration (section 6.3).

Modelling

The second-order HMM configurations investigated were:

- X2.
For this configuration, the trained first-order SV HMMs from the X1_GMM_CSVM HMM configuration were converted to second-order HMMs. The SV HMMs were retrained prior to classification.
- X2_CSVM.
The trained first-order CSVM HMM of configuration X1_GMM_CSVM was converted to a second-order HMM and retrained on the features of all the SVs. The SV HMMs were initialised from this trained second-order CSVM HMM and retrained prior to classification.
- C2.
For this configuration, the trained first-order SV HMMs from the X1_GMM_CSVM HMM configuration were converted to second-order context-emphasised HMMs. The second-order context-emphasised SV HMMs were retrained prior to classification.
- C2_CSVM.
The trained first-order CSVM HMM of configuration X1_GMM_CSVM was converted to a second-order context-emphasised CSVM HMM. The second-order context-emphasised CSVM HMM was retrained on the features of all the SVs. The SV HMMs were initialised from the trained CSVM model and retrained prior to classification.
- D2.
The trained first-order SV HMMs from the X1_GMM_CSVM HMM configuration were converted to second-order duration-emphasised HMMs. The second-order duration-emphasised SV HMMs were retrained prior to classification.
- D2_CSVM.
The trained first-order CSVM HMM of configuration X1_GMM_CSVM was converted to a second-order duration-emphasised CSVM HMM. The second-order duration-emphasised CSVM HMM was retrained on the features of all the SVs. The SV HMMs were initialised from the trained CSVM model and retrained prior to classification.

Training of all HMMs consisted of five iterations of the Viterbi algorithm.

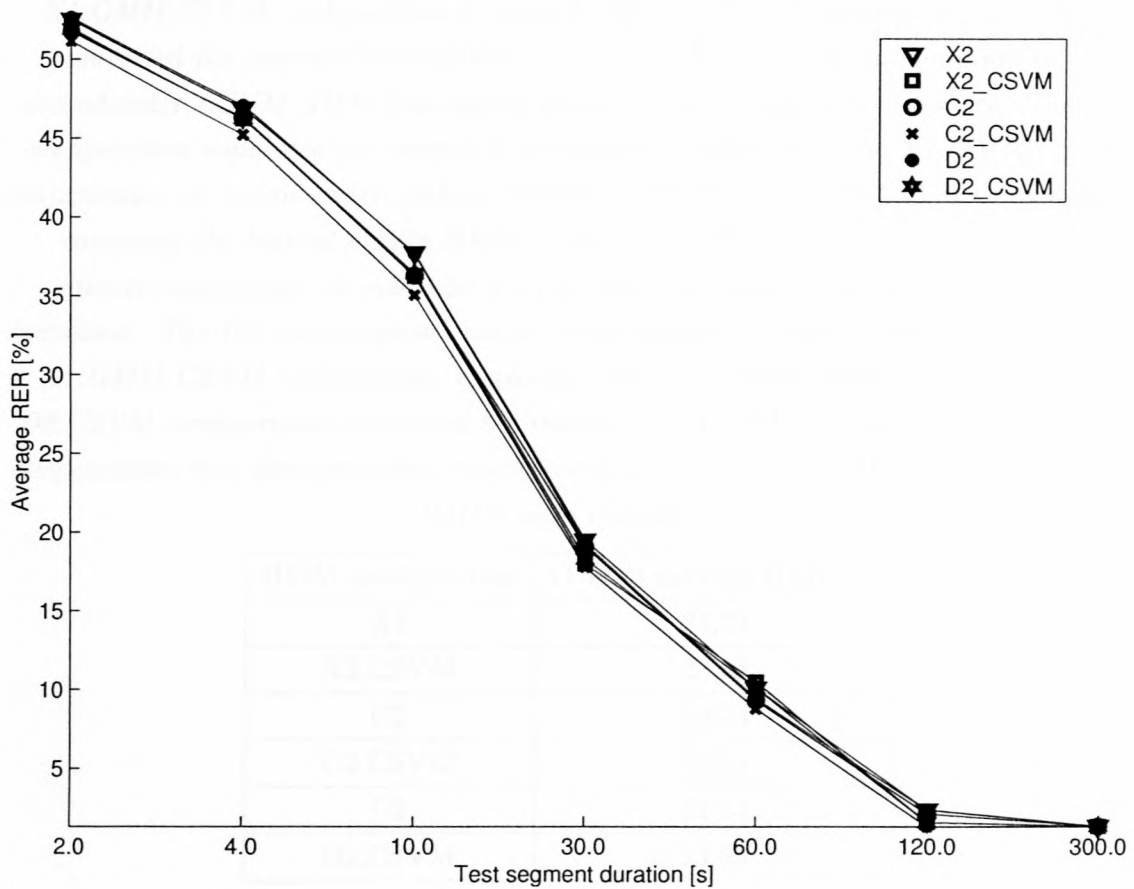


Figure 6.10: Average RER [%] for the second-order HMM configurations investigated. Configuration X2 converted the trained first-order SV HMMs from the X1_GMM_CSVM configuration to second-order models. Configuration X2_CSVM converted the trained CSVM HMM of the X1_GMM_CSVM configuration to a second-order CSVM HMM from which the SV HMMs were then initialised. The C2 configuration converted the trained first-order SV HMMs from the X1_GMM_CSVM configuration to second-order context-emphasised HMMs. The C2_CSVM configuration converted the trained CSVM HMM of the X1_GMM_CSVM configuration to a context-emphasised second-order CSVM HMM from which the SV HMMs were initialised. The D2 configuration converted the trained first-order SV HMMs from the X1_GMM_CSVM configuration to second-order duration-emphasised HMMs. The D2_CSVM configuration converted the trained CSVM HMM of the X1_GMM_CSVM configuration to a duration-emphasised second-order CSVM HMM from which the SV HMMs were initialised.

Table 6.6: Overall average RER [%] for the second-order HMM configurations investigated. Configuration X2 converted the trained first-order SV HMMs from the X1_GMM_CSVM configuration to second-order models. Configuration X2_CSVM converted the trained CSVM HMM of the X1_GMM_CSVM configuration to a second-order CSVM HMM from which the SV HMMs were then initialised. The C2 configuration converted the trained first-order SV HMMs from the X1_GMM_CSVM configuration to second-order context-emphasised HMMs. The C2_CSVM configuration converted the trained CSVM HMM of the X1_GMM_CSVM configuration to a context-emphasised second-order CSVM HMM from which the SV HMMs were initialised. The D2 configuration converted the trained first-order SV HMMs from the X1_GMM_CSVM configuration to second-order duration-emphasised HMMs. The D2_CSVM configuration converted the trained CSVM HMM of the X1_GMM_CSVM configuration to a duration-emphasised second-order CSVM HMM from which the SV HMMs were initialised.

HMM configuration	Overall average RER [%]
X2	24,34
X2_CSVM	23,68
C2	23,74
C2_CSVM	22,94
D2	24,24
D2_CSVM	23,69

Results

The detail classification results for the second-order HMM configurations are found in Table B.23-Table B.28 and Figure B.23-Figure B.28.

Interpretation

From Figure 6.10 and perhaps more clearly from Table 6.6, it is seen that the C2_CSVM second-order HMM configuration achieved the lowest overall average RER of the second-order HMM configurations investigated.

From the results presented in Table 6.6, it is clear that the HMM configurations which utilised a second-order CSVM HMM (X2_CSVM, C2_CSVM and D2_CSVM) fared better than the configurations which converted the SV HMMs to second-order HMMs (X2, C2 and D2). This result again highlights the benefit of utilising a CSVM to initialise the SV models, be it for GMM modelling or HMM modelling of the SVs.

Figure 6.11 compares the average RER attained by first-order HMM modelling (X1_GMM_CSVM configuration) and second-order HMM modelling (C2_CSVM configuration) of the SVs. The C2_CSVM configuration lowered the RER attained by the X1_GMM_CSVM configuration with 4,22%, from 23,95% to 22,94%. Both the first-order

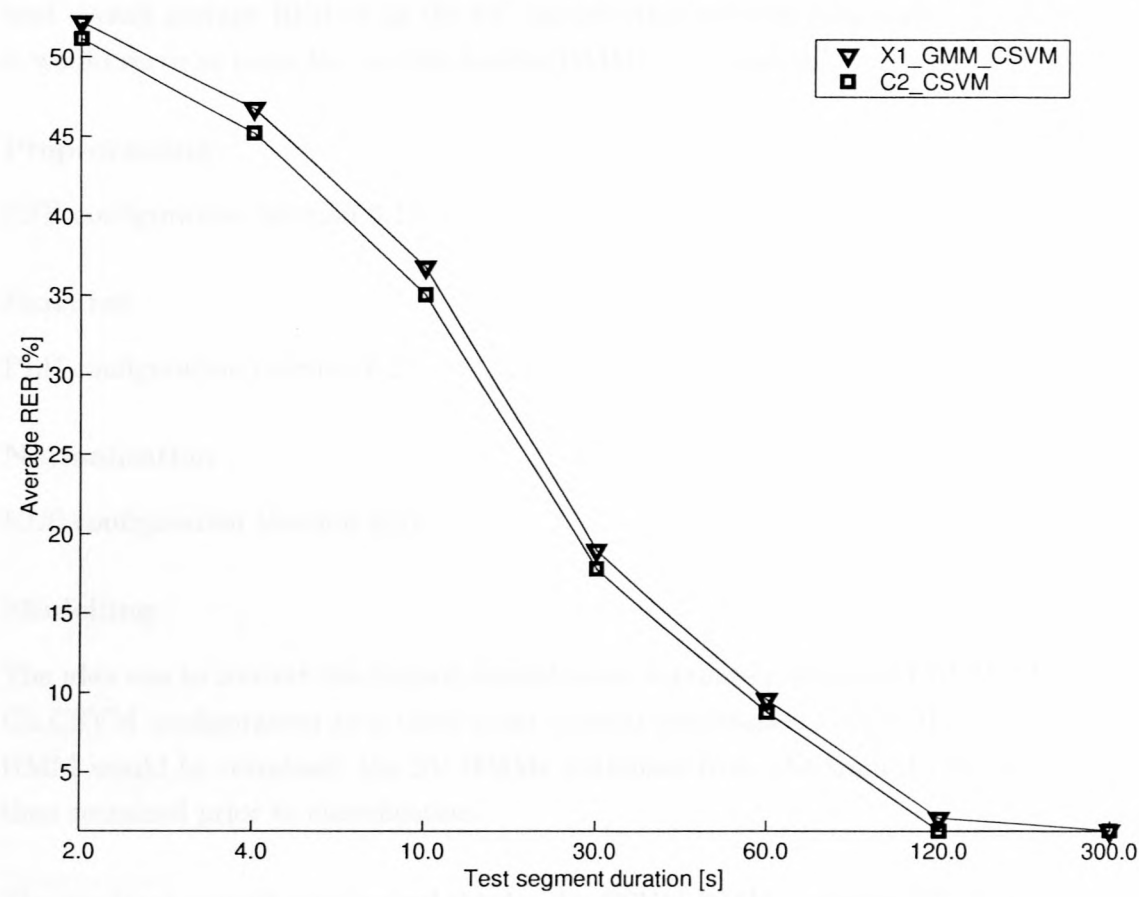


Figure 6.11: Average RER [%] attained by first-order HMM modelling (X1_GMM_CSVM configuration) and second-order HMM modelling (C2_CSVM configuration) of the SVs.

HMM configuration X1_GMM_CSVM and the second-order configuration C2_CSVM converge to the same average RER (1,33%) when the test segment duration is increased to 300s. This convergence in the average RER is attributed to possible data scarcity problems. The average RER measured around 300s is therefore not considered to be reliable.

Since the increase in HMM order resulted in a lowering of the overall average RER, it was decided to next investigate a third-order HMM configuration based on the C2_CSVM configuration.

6.4.6 Investigating a third-order HMM configuration for SV modelling

This experiment investigated the effect on the overall average RER if the HMM order was increased from second-order to third-order. Since the C2_CSVM configuration yielded the best overall average RER of all the SV classification systems considered up to this point, it would serve as basis for the third-order HMM configuration.

Preprocessing

PFP configuration (section 6.1).

Features

PLP configuration (section 6.2).

Normalisation

KLT configuration (section 6.3).

Modelling

The idea was to convert the trained second-order context-emphasised CSVM HMM of the C2_CSVM configuration to a third-order context-emphasised CSVM HMM. The CSVM HMM would be retrained; the SV HMMs initialised from the trained CSVM HMM and then retrained prior to classification.

The resultant context-emphasised third-order CSVM HMM contained 677 798 states with a total of 1 849 387 links. Parameter estimation of this model required more computational resources than the simulation PC had available.

Therefore, it was decided to utilise the second-best second-order HMM configuration (X2_CSVM) as basis for a third-order HMM configuration. The third-order HMM configuration was labelled X3.

Results

Figure 6.12 compares the average RER attained with second-order HMM SV modelling (X2_CSVM configuration) to that attained with third-order HMM SV modelling (X3 configuration). The detailed classification results attained with the X3 configuration are shown in Table B.29 and Figure B.29.

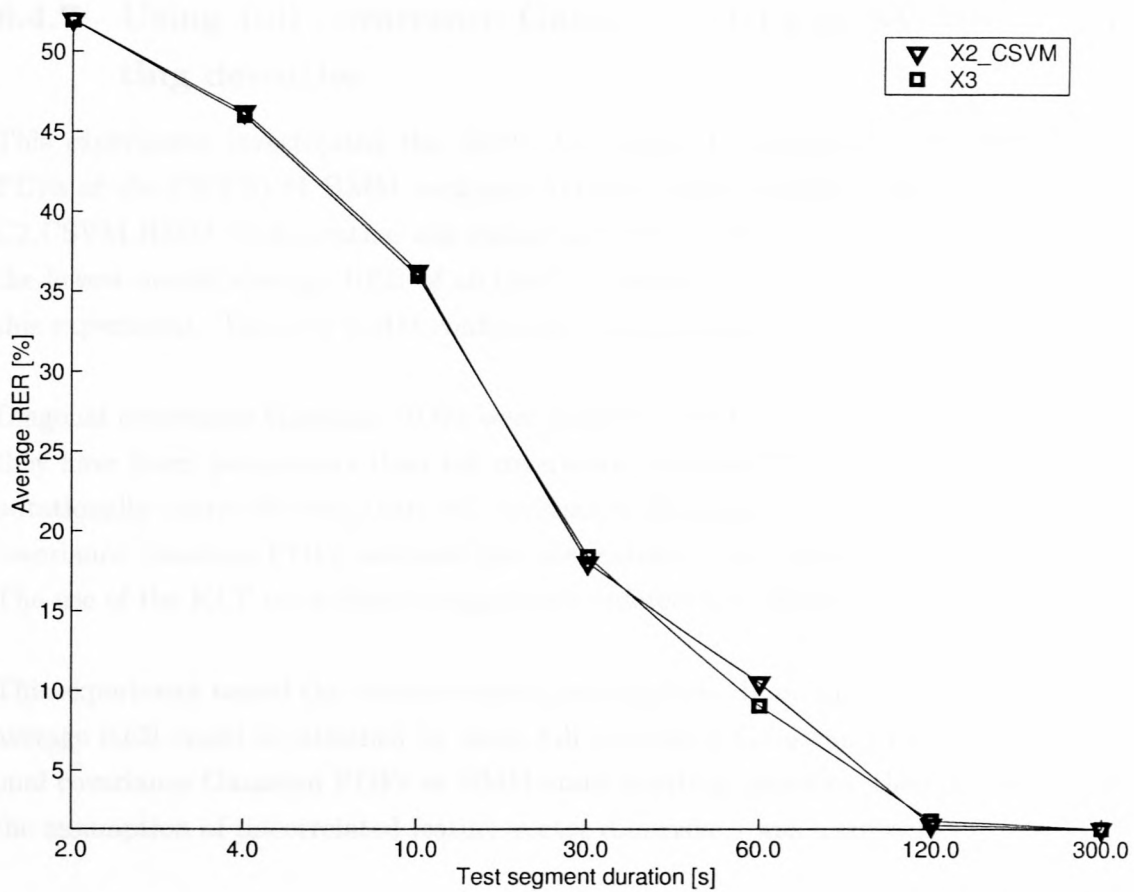


Figure 6.12: Average RER [%] attained by second-order HMM modelling (*X2_CSVM* configuration) and third-order HMM modelling (*X3* configuration) of the SVs.

Interpretation

The X3 configuration overall average RER of 23,49% was higher than the overall average RER (22,94%) of the best second-order HMM configuration (C2_CSVM) and it was only a 0,84% improvement on the RER (23,68%) of the second-order HMM configuration (X2_CSVM) which served as basis for the X3 configuration.

The low reduction (with respect to the X2_CSVM configuration) in overall average RER attained by the X3 system indicated that the SV HMMs did not benefit from the new information provided by the increase in HMM order.

6.4.7 Using full covariance Gaussian PDFs as SV HMM emitting densities

This experiment investigated the effect of replacing the diagonal covariance Gaussian PDFs of the C2_CSVM HMM configuration with full covariance Gaussian PDFs. The C2_CSVM HMM configuration was chosen as basis for this experiment, since it attained the lowest overall average RER of all the SV classification configurations tested prior to this experiment. The new HMM configuration was labelled C2_FC.

Diagonal covariance Gaussian PDFs were utilised in all the preceding experiments, since they have fewer parameters than full covariance Gaussian PDFs and are therefore computationally more efficient than full covariance Gaussian PDFs. The use of diagonal covariance Gaussian PDFs assumes that the feature vector dimensions are uncorrelated. The use of the KLT normaliser configuration enforces this assumption.

This experiment tested the aforementioned assumptions. If an improvement in the overall average RER could be attained by using full covariance Gaussian PDFs instead of diagonal covariance Gaussian PDFs as HMM state emitting densities, then it might be that the assumption of uncorrelated feature vector dimensions was incorrect.

Preprocessing

PFP configuration (section 6.1).

Features

PLP configuration (section 6.2).

Normalisation

KLT configuration (section 6.3).

Modelling

The C2_FC configuration was created in exactly the same manner as the C2_CSVM configuration, but it utilised full covariance Gaussian PDFs wherever the C2_CSVM configuration utilised diagonal covariance Gaussian PDFs. The procedure utilised to create the C2_CSVM configuration, and therefore the C2_FC configuration, has been described in subsection 6.4.3.

Results

Figure 6.13 compares the results obtained by using full covariance Gaussian PDFs as HMM emitting densities (C2_FC configuration) instead of diagonal covariance Gaussian PDFs (C2_CSVM configuration). The detailed classification results attained by the C2_FC

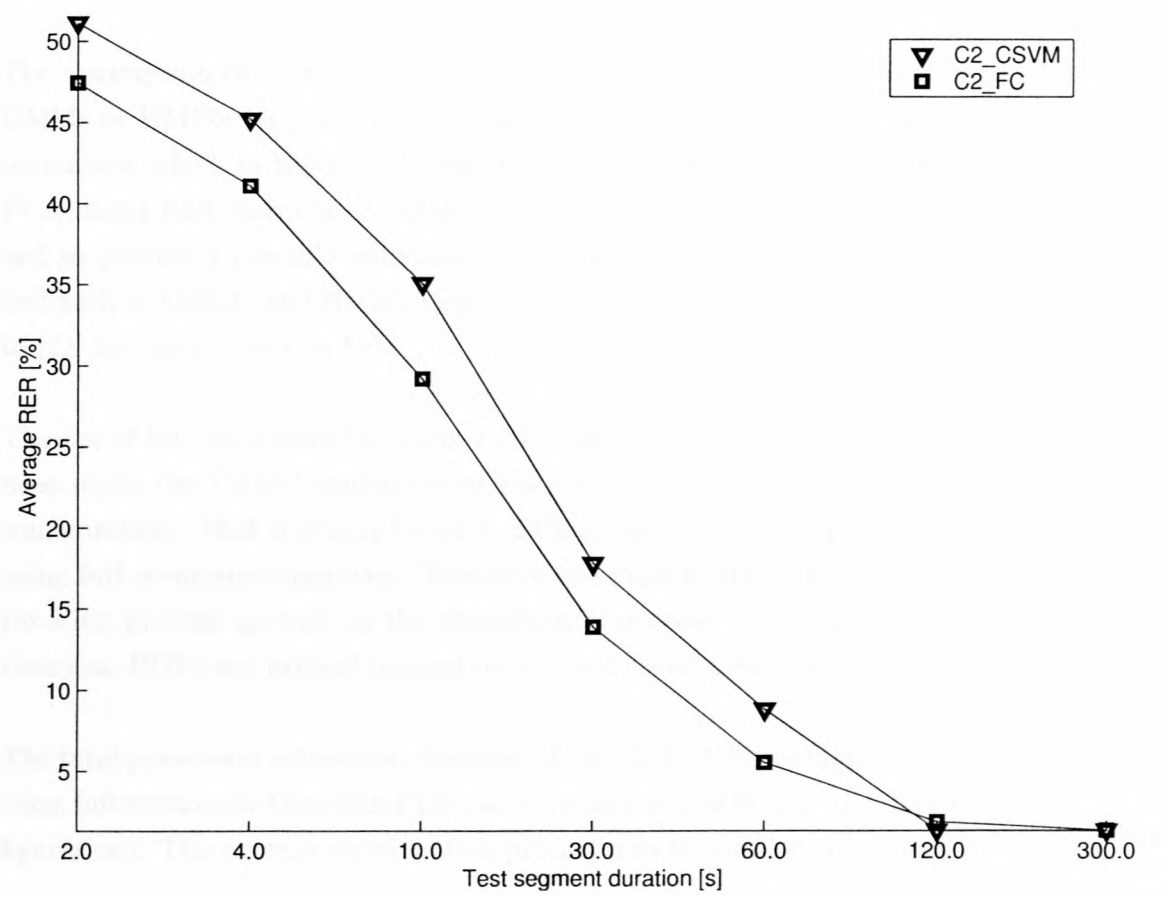


Figure 6.13: Comparison of the average RER [%] attained by the C2_CSVM HMM configuration (diagonal covariance Gaussian PDFs) and the C2_FC HMM configuration (full covariance Gaussian PDFs).

configuration are shown in Table B.30 and Figure B.30.

6.3 Comparison of SV classification systems

Interpretation

From the average RER figures presented in Figure 6.13 it is clear that the C2_FC configuration improved the overall average RER attained by the C2_CSVM configuration. The overall average RER was improved by 12,68%, from 22,94% to 20,03%. The crossover in Figure 6.13 in the vicinity of 120s is attributed to data scarcity problems. The average RER measured beyond 120s is therefore considered to be unreliable.

The improvement in overall average RER, attained by replacing the diagonal covariance Gaussian PDFs of the C2_CSVM configuration with full covariance Gaussian PDFs indicates that the assumption of uncorrelated feature vector dimensions was wrong. This implies that the motivation for using the KLT normaliser is no longer valid. In fact, it is inappropriate to use the KLT normaliser with GMM- and HMM pattern modelling.

The assumption of uncorrelated feature vector dimensions is no longer valid when using GMMs or HMMs for pattern modelling. It would have been more appropriate to use a normaliser which is based on Linear Discriminant Analysis (LDA) instead of using the PCA-based KLT normaliser. LDA is used to improve inter-class (category) separability and to provide a possible reduction in feature vector dimension. Statistical pattern models such as GMMs and HMMs benefit from the class separation provided by LDA. Refer to [22] for more detail on LDA.

The use of full covariance Gaussian PDFs instead of diagonal covariance Gaussian PDFs does make the C2_FC configuration less computationally efficient than the C2_CSVM configuration. That is simply because of the greater number of parameters involved when using full covariance matrices. Therefore one expects the duration of the parameter estimation process as well as the classification process to increase when full covariance Gaussian PDFs are utilised instead of diagonal covariance Gaussian PDFs.

The total parameter estimation duration of the C2_CSVM configuration increased by 18% when full covariance Gaussian PDFs were utilised as HMM emitting densities (C2_FC configuration). The average classification process duration was increased by 22%.

The average classification duration was obtained by measuring the duration of the classification process for each of the test segments utilised and taking the average of these durations. The issue of the SV classification system computational efficiency is addressed in the following section.

6.5 Comparison of SV classification systems

Throughout the experimental investigation, we have compared the SV classification systems in terms of the overall average RER attained by each. It seems logical that for a SV classification system one would first try to minimise the overall average RER of the system before paying attention to other concerns.

One such concern is that of computational efficiency of the SV classification systems.

The computational efficiency of a SV classification system will determine whether it is usable in real-time speech recognition applications.

If a SV classification system has a very large delay between when it receives speech input and when it produces a classification decision, then it will not be useful in real-time speech recognition applications. This will be true even if it is capable of very accurate classification.

The delay between the input of speech to the SV classification system and the output of a decision by the system will be called the ‘processing delay’ of the system. The processing delay will largely be determined by the duration of the feature extraction-, model parameter estimation- and classification process durations.

Of the aforementioned processing stages, the model parameter estimation stage is often computationally the most demanding, since it has to process a significant duration of training speech; and do it numerous times if an iterative training process like the EM-algorithm is employed.

Therefore a major portion of the processing delay for a SV classification system will be the model parameter estimation duration. If the SV classification system model parameters are estimated offline, then the duration of the model parameter estimation process is no longer a concern. For such an application; only the duration of extracting the features used in classification, as well as the duration of the classification process will be of importance.

For each of the experiments conducted, the durations of the feature extraction-, parameter estimation- and classification process were recorded. For each of the major SV classification systems investigated, the sum of the configuration’s feature extraction process- and classification process durations will be presented.

The sum of these durations may then be compared against the total duration of the test utterances utilised in the classification experiments, in order to determine the real-time performance of the SV classification configurations. Note that the duration of the feature extraction process will only be the duration of extracting the features used for testing (classification).

In order for the processing durations to have meaning, the computer setup used to conduct the experiments should be described briefly. All experiments were conducted on a

dual CPU 1,26 GHz Intel Pentium 3 Tualatin PC with 1GB RAM and Red Hat Linux 7.3 as operating system.

The durations measured for the feature extraction- and classification processes were measured as if all processing took place serially, on one of the PC's CPUs. I.e. all of the SV classification processing stages utilised the PC CPUs in parallel, but the processing durations were recorded separately for each process that was executed.

The experiments were conducted using the pattern recognition software library PatrecII. This library is written in C++; with numerous members of the Digital Signal Processing (DSP) group of the Faculty of Electronic Engineering, Stellenbosch University, contributing code to it over the years.

6.5.1 Overall average RER attained by the SV classification systems

We first look at the overall average RER attained by the SV classification systems, since this is the primary indicator of a SV classification system's performance.

Table 6.7 shows the overall average RER for the major SV classification systems as well as the average RER attained by each system on all of the test segment durations. The SV classification systems represented in Table 6.7 were the systems which attained the lowest overall average RER of all the systems investigated in the experiment they formed part of. Also shown in Table 6.7 are the lower and upper limits of the 95% confidence interval for each of the average RER values computed. The confidence intervals were computed based on the assumption that each average RER value was obtained from independent Bernoulli trials.

The initial best overall average RER of 52,05%, attained by the PFP system, was systematically decreased as the experiments progressed and the SV classification system was refined. Eventually, an overall average RER of 20,03% was attained by the C2_FC system. This constitutes a 61,52% decrease in the overall average RER attained by the PFP system.

The systematic decrease in the overall average RER is better illustrated by Figure 6.14. This figure shows the average RER for each of the major SV classification systems, as attained on each of the test segment durations. Also shown in this figure are error bars which indicate the 95% confidence interval for each of the average RER values.

As indicated by Table 6.7 and Figure 6.14, the C2_FC system attained the lowest overall

Table 6.7: Overall average RER [%] for the major SV classification systems investigated as well as the average RER attained on each test segment duration. Also shown are the 95% confidence intervals (in square brackets) for the average RER values, assuming independent Bernoulli trials.

System and overall average RER [%]	Average RER [%] vs. test segment duration [s]						
	2s	4s	10s	30s	60s	120s	300s
PFP 52,05%	[68,81] 68,31% [67,8]	[66,82] 66,1% [65,37]	[63,51] 62,34% [61,16]	[56,57] 54,48% [52,38]	[53,82] 50,85% [47,88]	[43,13] 38,95% [34,92]	[29,46] 23,33% [18,16]
PLP 45,86%	[65,62] 65,11% [64,59]	[63,17] 62,43% [61,69]	[59,46] 58,27% [57,06]	[51,48] 49,38% [47,28]	[42,92] 39,97% [37,1]	[35,89] 31,84% [28,05]	[19,3] 14,0% [9,98]
KLT 42,93%	[64,67] 64,16% [63,63]	[62,1] 61,35% [60,6]	[56,88] 55,67% [54,47]	[48,41] 46,3% [44,22]	[38,28] 35,39% [32,6]	[30,19] 26,32% [22,78]	[16,29] 11,33% [7,74]
GMM_70 41,58%	[61,52] 60,99% [60,46]	[59,55] 58,8% [58,04]	[54,27] 53,06% [51,85]	[44,73] 42,64% [40,58]	[36,55] 33,68% [30,93]	[31,83] 27,89% [24,28]	[19,3] 14,0% [9,98]
GMM_CSVM 35,44%	[58,24] 57,7% [57,17]	[55,46] 54,7% [53,93]	[48,74] 47,53% [46,32]	[35,9] 33,88% [31,92]	[28,51] 25,82% [23,3]	[21,91] 18,42% [15,38]	[14,76] 10,0% [6,65]
X1_GMM_CSVM 23,95%	[52,7] 52,16% [51,61]	[47,5] 46,73% [45,96]	[37,97] 36,79% [35,63]	[20,67] 18,97% [17,37]	[11,46] 9,57% [7,96]	[3,7] 2,11% [1,19]	[3,95] 1,33% [0,44]
C2_CSVM 22,94%	[51,65] 51,11% [50,57]	[45,97] 45,21% [44,44]	[36,21] 35,05% [33,9]	[19,45] 17,79% [16,24]	[10,62] 8,78% [7,24]	[2,68] 1,32% [0,64]	[3,95] 1,33% [0,44]
C2_FC 20,03%	[47,96] 47,42% [46,88]	[41,83] 41,08% [40,32]	[30,3] 29,19% [28,1]	[15,38] 13,87% [12,48]	[7,03] 5,5% [4,3]	[3,36] 1,84% [1,0]	[3,95] 1,33% [0,44]

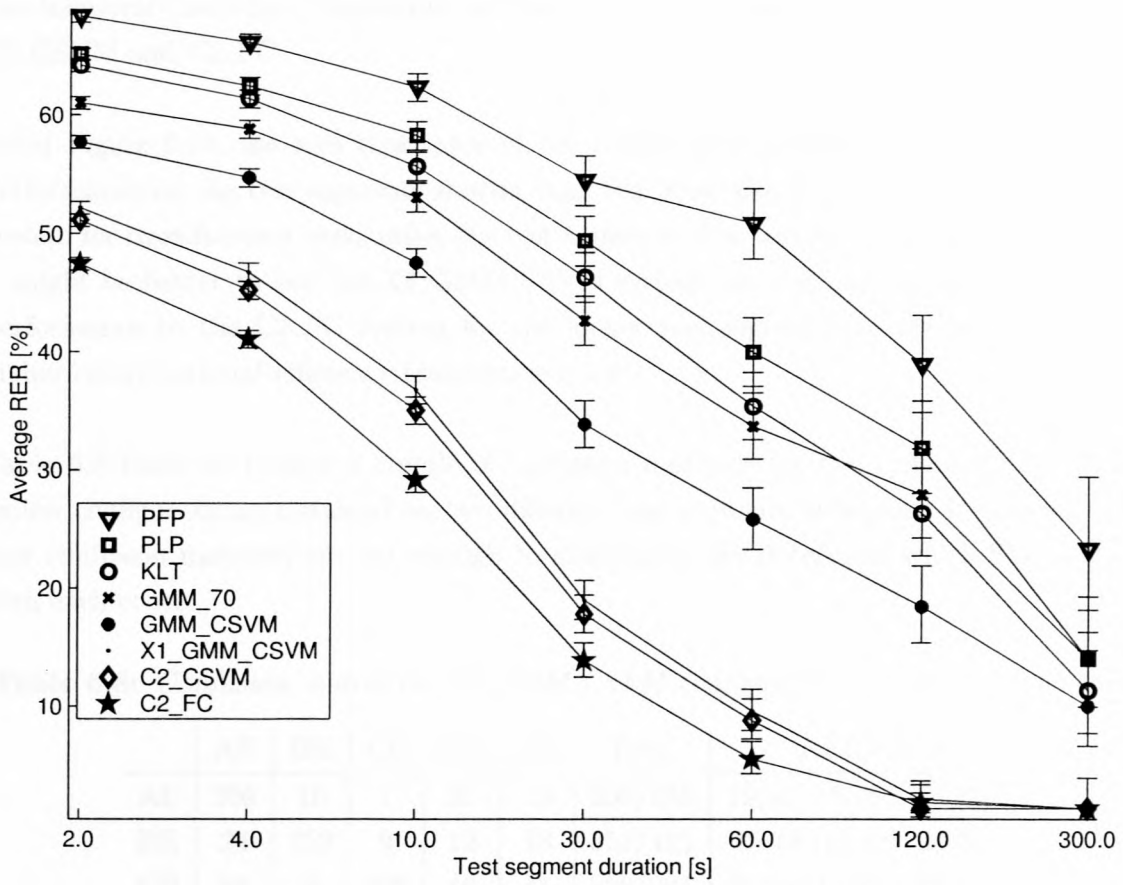


Figure 6.14: Average RER [%] for the major SV classification systems investigated. Also shown are the 95% confidence intervals (indicated by the error bars) for the average RER values, assuming independent Bernoulli trials.

average RER of the SV classification systems investigated. Its overall average RER of 20,3% may appear relatively high, but it should be remembered that this overall average RER was computed over all the test segment durations utilised. These segments varied in length from as little as 2s to 300s (Table 6.1).

From Table 6.7 and Figure 6.14 one sees that the X1_GMM_CSVM, C2_CSVM and C2_FC average RER figures drop below 10% for test segments longer than 60s. This indicates that further research is required to enhance the SV classification system performance for test segments shorter than 60s.

It is postulated that the system performance on the shorter test segments will be improved not by utilising more complicated SV modelling techniques, but by improving the speech preprocessing-, feature extraction- and feature normalisation configurations utilised in the thesis. That is because the shorter test segments do not necessarily utilise

the temporal modelling capabilities of the SV classification systems X1_GMM_CSVM, C2_CSVM and C2_FC.

From Figure 6.14 one sees that none of the classification systems achieve satisfactory performance on the test segments shorter than 60s. One also sees that C2_FC is the best system for classification tasks using 60s test segments. For test segments longer than 60s, it might be better to use the X1_GMM_CSVM system since it has similar classification performance to the C2_FC system for the longer test segments, while having a much higher computational efficiency (subsection 6.5.2).

Table 6.8-Table 6.11 show a couple of confusion matrices for two different SV classification configurations, obtained on two different test segment durations. It appears as if four confusion matrices are not enough to unambiguously determine which SVs confuse with each other.

Table 6.8: *Confusion matrix for X1_GMM_CSVM configuration on 30s test segments.*

	AE	BE	CE	EE	IE	Total	RER [%]
AE	206	10	7	20	13	206/256	19,53 [15,78 : 23,92]
BE	24	352	9	12	18	352/415	15,18 [12,51 : 18,3]
CE	28	8	228	16	11	228/291	21,65 [17,95 : 25,87]
EE	23	9	3	205	38	205/278	26,26 [22,16 : 30,81]
IE	6	8	3	24	248	248/289	14,19 [11,14 : 17,9]

Table 6.9: *Confusion matrix for X1_GMM_CSVM configuration on 120s test segments.*

	AE	BE	CE	EE	IE	Total	RER [%]
AE	63	0	0	1	0	63/64	1,56 [0,35 : 6,71]
BE	0	102	0	1	0	102/103	0,97 [0,22 : 4,24]
CE	1	0	71	0	0	71/72	1,39 [0,31 : 5,99]
EE	0	0	0	64	5	64/69	7,25 [3,57 : 14,15]
IE	0	0	0	0	72	72/72	0,0 [0,0 : 3,62]

Table 6.10: *Confusion matrix for C2_FC configuration on 30s test segments.*

	AE	BE	CE	EE	IE	Total	RER [%]
AE	215	6	10	15	10	215/256	16,02 [12,6 : 20,14]
BE	11	365	11	17	11	365/415	12,05 [9,66 : 14,93]
CE	13	5	256	7	10	256/291	12,03 [9,24 : 15,52]
EE	24	10	4	215	25	215/278	22,66 [18,81 : 27,04]
IE	5	5	4	9	266	266/289	7,96 [5,71 : 10,98]

Table 6.11: *Confusion matrix for C2_FC configuration on 120s test segments.*

	AE	BE	CE	EE	IE	Total	RER [%]
AE	64	0	0	0	0	64/64	0,0 [0,0 : 4,06]
BE	0	100	1	2	0	100/103	2,91 [1,17 : 7,07]
CE	0	0	72	0	0	72/72	0,0 [0,0 : 3,62]
EE	0	1	1	65	2	65/69	5,8 [2,63 : 12,3]
IE	0	0	0	0	72	72/72	0,0 [0,0 : 3,62]

6.5.2 Computational efficiency of the SV classification systems

Next we compare the SV classification systems in terms of their computational efficiency, which is measured by the SV classification system processing delay. As discussed in the beginning of this section, the processing delay will be measured in terms of the average classification process duration and the duration of the feature extraction process (for the classification features).

The average classification process duration is obtained by measuring the duration of the classification process for each of the test segment lengths utilised, and then taking the average of these measured durations. Table 6.12 shows the duration of extracting the test features, the average classification duration and the sum of these durations (‘Processing delay’) for the major SV classification systems investigated in the thesis. The ‘Processing delay’ listed in Table 6.12 will be compared against the total duration of the test utterances used in the classification experiments. The duration of the test utterances was twelve hours, forty-seven minutes and twenty-two seconds (Table A.2).

All the SV classification configurations listed in Table 6.12, except for the PFP system, utilised the nine-dimensional (eight-dimensional PLP plus frame energy) feature representation of the PLP system. Feature extraction from the test utterances took five minutes

Table 6.12: *Processing delay for the major SV classification systems investigated.*

System	Test features extraction duration	Average classification duration	Processing delay
PFP	13m 56s	12s	14m 8s
PLP	5m 46s	9s	5m 55s
KLT	5m 46s	33s	6m 19s
GMM_70	5m 46s	14m 37s	20m 23s
GMM_CSVM	5m 46s	14m 36s	20m 22s
X1_GMM_CSVM	5m 46s	22m 29s	28m 15s
C2_CSVM	5m 46s	20h 28m 50s	20h 34m 36s
C2_C	5m 46s	25h 9m 19s	25h 15m 5s

and forty-six seconds for the PLP system. For the MFCC18/PFP system, which utilised eighteen-dimensional MFCCs, extracting the test utterance features took thirteen minutes and fifty-six seconds.

With this in mind, one may see from Table 6.12 that all the major SV classification systems, except for the second-order HMM SV classification systems C2_CSVM and C2_FC, are computationally efficient. Only the C2_CSVM and C2_FC classification systems have processing delays which exceed the duration of the test utterances.

In order to illustrate the trade-off between classification accuracy (overall average RER) and computational efficiency, let us compare the X1_GMM_CSVM and C2_FC systems. The C2_FC system processing delay is 1,97 times the duration of the speech utilised for testing. On the other hand, the X1_GMM_CSVM system processing delay is 0,0355 times the duration of the speech utilised for classification. For the C2_FC system, the test features extraction process duration comprises a mere 0,34% of this system's processing delay. For the X1_GMM_CSVM system, the duration of extracting the test features comprises 18,65% of this system's processing delay.

The C2_FC system has an overall average RER of 20,03% which is 16,37% lower than that of the X1_GMM_CSVM system overall average RER of 23,95%. This 16,37% decrease in overall average RER comes at a 5459,9% increase in processing delay. This result highlights the importance of the accuracy (overall average RER) vs. processing delay trade-off. The SV classification system chosen for a particular speech recognition application will be determined by the accuracy requirements as well as the processing delay requirements of the application.

From the previous discussion, it is seen that the C2_FC system is not yet suitable for deployment in speech recognition applications which require real-time speech processing performance. The simplest (but expensive) solution for reducing the processing delay of the C2_FC system would be to utilise a PC with a higher CPU clock-speed. Ideally one would like to investigate the optimisation of the algorithms (and software in general) utilised in the SV classification speech processing. This issue is probably constantly being investigated by speech recognition practitioners all over the world.

A large reduction in the processing delay of the SV classification system would be obtained if much of the speech processing could be parallelised. The degree to which this is possible will largely depend on the nature of the algorithms employed in the speech recognition software.

Many of the algorithms utilised in current speech recognition applications might be difficult to parallelise. Also, parallelisation of the speech processing algorithms and software will only be useful on multi-CPU computer systems.

Chapter summary

In this chapter we presented the experimental investigation for the thesis. The purpose of these experiments were to develop a SV classification system with as low an overall average RER as possible.

This was accomplished by investigating the different processing stages of the SV classification system in turn. For each of these stages, we utilised a number of different configurations and determined which of these configurations produced the lowest overall average RER. For each of the experiments, we described the experimental setup used, provided the experimental results and interpreted the experimental results.

The best-performing configuration of each stage was incorporated in subsequent experiments. In this manner we managed to reduce the initial best overall average RER of 52,05% to 20,03%. We also compared the major SV classification systems developed in this thesis in terms of their overall average RER as well as their computational efficiency.

The following chapter concludes the thesis with a summary of the key experimental results, a number of suggestions for improving the SV classification system and a summary of the thesis achievements.

Chapter 7

Conclusion and recommendations

Introduction

In this chapter the key experimental results are presented along with a number of recommendations for improving the SAE SV classification system performance. This is followed by the thesis conclusion.

7.1 Key experimental results

- The experiments presented in chapter 6 showed that classification of the SAE SVs is possible, using a transcription-less speech recognition approach.
- It was shown that it is possible to systematically decrease the system overall average RER from its initial value of 52,05% to the eventual value of 20,03% (Figure 6.14).
- The classification results of section 6.1 showed that use of the PFP improves the SV classification system robustness against non-speech events.
- The experiment of section 6.2 showed that PLP cepstra outperform MFCCs on the SAE SV classification task (Figure 6.2).
- The benefit of using a global initialisation model (CSVM) for initialising the SV model parameters prior to parameter estimation with the EM-algorithm was showed by the experimental results of subsection 6.4.2, subsection 6.4.3 and subsection 6.4.5.
- It was shown that HMMs outperform GMMs on the SAE SV classification task (Figure 6.8 and Table 6.7).
- The experiment of subsection 6.4.7 showed that a reduction in overall average RER is attained when the diagonal covariance Gaussian PDFs of the HMMs are replaced with full covariance Gaussian PDFs (Figure 6.13).

- The discussion in subsection 6.5.1 and subsection 6.5.2 highlighted the trade-off between classification accuracy and computational efficiency.
- From the comparison of the overall average RER and average RER figures for the SV classification systems (Table 6.7 and Figure 6.14), it appears as if C2_FC is the best system for classification tasks using 60s test segments. For test segments longer than 60s, it might be better to use the X1_GMM_CSVM system since it has similar classification performance to the C2_FC system for the longer test segments, while having a much higher computational efficiency.

7.2 Future work

There is still much work to be done with regards to SAE SV classification and speech recognition using South African languages in general. For the classification of SAE SVs, the following could be investigated in future research efforts:

- Automatic parameter estimation for the PFP preprocessor.
Currently the PFP parameters are deduced from manual inspection of a number of speech files from the target speech corpus. It would be convenient if these parameters could be estimated automatically.
- Alternative speech feature extraction configurations.
Speech feature extraction configurations, other than those used in this thesis, could be investigated. It might be possible to find a feature representation which provides better parameterisation of the SV spectral characteristics than the PLP cepstra utilised in this thesis. It was postulated in subsection 6.5.1 that better parameterisation of the SV spectral characteristics will result in better SV classification performance on the test segments which are shorter than 60s in duration.
- Investigate the use of different feature extraction frame lengths.
Throughout this thesis a feature extraction frame length of 20ms was utilised, with a frame skip of 10ms. Future research might investigate the use of different frame length- and frame skip values. The trade-off between spectral- and temporal resolution of the speech features and the effect on the overall SV RER could be investigated.
- Alternative feature normalisation techniques may be investigated.
A very basic selection of feature normalisation configurations were investigated in this thesis. Future research should definitely look into the use of LDA-based normalisers for use with the GMM and HMM pattern-modelling configurations (subsection 6.4.7). The PCA-based KLT normaliser was inadvertently used with GMM

and HMM pattern-modelling configurations for which the assumption of uncorrelated feature vector dimensions is no longer valid.

- The optimisation of the speech recognition algorithms employed in the thesis should be investigated.

This is considered to be important to speech recognition research in general and not just the SV classification problem.

- Investigate the use of speech transcriptions in a SV classification system.

This will be possible once the AST SAE corpus speech transcription process has been finalised. One may then be able to investigate transcription-dependent pattern modelling techniques such as the phoneme recogniser/N-gram language model combination.

- Investigate the use of separate SV models which are gender- and telephone channel specific (section 3.2).

The experimental investigation results showed a significant difference between the RER figures for the different SVs, especially with the shorter test segment durations. This was attributed to different speaker gender- and telephone type characteristics for the SVs. Once the AST SAE corpus has been finalised it might be worthwhile to investigate whether gender- and telephone type specific SV models would be able to improve the SV classification performance.

7.3 Conclusion

After reviewing the thesis work, the conclusion is reached that the thesis objectives (section 1.2) were met. A speech recognition system capable of classifying the SAE SVs with an overall average RER of 20,03% was developed. This speech recognition system does not require speech transcriptions. The overall average RER of this system drops below 10% if test segments longer than 60s are utilised.

This SV classification system was developed in a logical and systematic manner. In the process of developing this SV classification system, a couple of novel pattern recognition and speech processing techniques were developed. These are the CARP- and PFP preprocessors and the HVQ vector quantisation technique.

The shortcomings of the developed SV classification system were identified and highlighted in section 6.5. These shortcomings are the high overall average RER for test segments shorter than 60s (subsection 6.5.1) and the computational inefficiency of the system (subsection 6.5.2).

Further research is required to reduce the high overall RER of the system with test segment durations of less than 60s; as well as to address the computational inefficiency of the SV classification system.

Bibliography

- [1] "AST project website", URL: <http://www.ast.ac.za>, accessed on 12/08/2010.
- [2] "Statistics South Africa Census 2007", 2007, <http://www.statssa.gov.za/SpecialPublications/Census2007/Rep45a.pdf>, accessed 12/08/2010.
- [3] *The Pocket Oxford Dictionary*, English edition, Oxford University Press, 1996.
- [4] ARSLAN, I. M. and HANSEN, J. H. L., "Frequency characteristics of natural and synthesized speech," in *Proceedings of the IEEE International Conference on Acoustics, Signal and Speech Processing*, vol. 2, p. 11-12, 1987.
- [5] ATAL, B. S. and HANAUER, S. L., "Speech analysis by prediction of the speech wave," *Journal of the Acoustical Society of America*, April 1971, Vol. 50, No. 2, pp. 637-655.
- [6] BARNARD, E. and YAN, Y., "An approach to automatic language identification based on language dependent phone recognition," in *Proceedings of the 1995 International Conference on Acoustics, Speech and Signal Processing*, pp. 3511-3514, 1995.
- [7] CHEN, T., HUANG, C., CHANG, F., and WANG, J., "Automatic word identification using Gaussian mixture models," in *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, 1-4 Oct 2004, pp. 1-5.
- [8] CRYSTAL, D., *The Cambridge encyclopedia of language*, Second edition, Cambridge University Press, 1997.
- [9] DELLER, J. R., HANSEN, J. H. L., and PROAKIS, J. G., *Discrete-Time Processing of Speech Signals*, Reprint edition, IEEE Press, 2000.
- [10] DEVLIVER, P. A. and KITTLER, J., *Pattern recognition: A statistical approach*, Prentice Hall International, 1982.
- [11] DU PRETZ, J. A. and WEBER, D. M., "Automatic language recognition using high-order HMMs," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, no. 1074, 1998.

Bibliography

- [1] "AST project website." URL: <http://espresso.ee.sun.ac.za/ast/>.
- [2] "Statistics South Africa Census 2001." URL: <http://www.statssa.gov.za/SpecialProjects/Census2001/digiAtlas/index.html>.
- [3] *The Pocket Oxford Dictionary*. Eighth edition. Oxford University Press, 1992.
- [4] ARSLAN, L. M. and HANSEN, J. H. L., "Frequency characteristics of foreign accented speech." in *Proceedings of the IEEE International Conference on Acoustics, Signal and Speech Processing*, vol. 2, p. 1123, 1997.
- [5] ATAL, B. S. and HANAUER, S. L., "Speech analysis and synthesis by linear prediction of the speech wave." *Journal of the Acoustical Society of America*, April 1971, Vol. 50, No. 2, pp. 637–655.
- [6] BARNARD, E. and YAN, Y., "An approach to automatic language identification based on language-dependent phone recognition." in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 5, pp. 3511–3514, 1995.
- [7] CHEN, T., HUANG, C., CHANG, E., and WANG, J., "Automatic accent identification using Gaussian mixture models." in *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU2001)*, 2001.
- [8] CRYSTAL, D., *The Cambridge encyclopedia of language*. Second edition. Cambridge University Press, 1997.
- [9] DELLER, J. R., HANSEN, J. H. L., and PROAKIS, J. G., *Discrete-Time Processing of Speech Signals*. Reprint edition. IEEE Press, 2000.
- [10] DEVIJVER, P. A. and KITTLER, J., *Pattern recognition : A statistical approach*. Prentice Hall International, 1982.
- [11] DU PREEZ, J. A. and WEBER, D. M., "Automatic language recognition using high-order HMMs." in *Proceedings of the IEEE International Conference on Spoken Language Processing*, no. 1074, 1998.

- [12] DU PREEZ, J. A. and WEBER, D. M., "Efficient high-order hidden Markov modelling." in *Proceedings of the IEEE International Conference on Spoken Language Processing*, no. 1073, 1998.
- [13] FISHER, W. M., DODDINGTON, G. R., and GOUDIE-MARSHALL, K. M., "The DARPA speech recognition research database: specifications and status." in *Proceedings of DARPA Workshop on Speech Recognition*, p. 93, 1986.
- [14] GONZALEZ, R. C. and WOODS, R. E., *Digital Image Processing*. Second edition. Prentice Hall, 2002.
- [15] HERMANSKY, H., "Perceptual linear predictive (PLP) analysis for speech." *Journal of the Acoustical Society of America*, 1990, Vol. 87, pp. 1738–1752.
- [16] HIERONYMUS, J. and KADAMBE, S., "Robust spoken language identification using large vocabulary speech recognition." in *Proceedings of the IEEE International Conference on Acoustics, Signal and Speech Processing*, vol. 2, p. 1111, 1997.
- [17] HOROWITZ, P. and HILL, W., *The art of electronics*. Second edition. Cambridge University Press, 1997.
- [18] HUANG, X., ACERO, A., and HON, H., *Spoken language processing: A guide to theory, algorithm and system development*. First edition. Prentice-Hall, 2001.
- [19] HUGGINS, A. W. F. and PATEL, Y., "The use of shibboleth words for automatically classifying speakers by dialect." in *Proceedings of the IEEE International Conference on Spoken Language Processing*, p. 1455, 1996.
- [20] JUANG, B. and RABINER, L., *Fundamentals of Speech Recognition*. First edition. Prentice-Hall, 1993.
- [21] KAT, L. W. and FUNG, P., "Fast accent identification and accented speech recognition." in *Proceedings of the IEEE International Conference on Acoustics, Signal and Speech Processing*, vol. 1, p. 2349, 1999.
- [22] KEINOSUKE, F., *Introduction to statistical pattern recognition*. Second edition. Academic Press, 1990.
- [23] KUMPF, K. and KING, R. W., "Automatic accent classification of foreign accented Australian English speech." in *Proceedings of the IEEE International Conference on Spoken Language Processing*, no. 1740, 1996.

- [24] LAY, D. C., *Linear algebra and its applications*. Second edition. Addison Wesley Longman, 1997.
- [25] LINCOLN, M., COX, S., and RINGLAND, S., "A comparison of two unsupervised approaches to accent identification." in *Proceedings of the IEEE International Conference on Spoken Language Processing*, no. 465, 1998.
- [26] MAKHOUL, J., ROUCOS, S., and GISH, H., "Vector quantization in speech coding." in *Proceedings of the IEEE*, vol. 73, pp. 1551–1588, 1985.
- [27] MALMKJAER, K. (Ed.), *The linguistics encyclopedia*. Second edition. Routledge, 2002.
- [28] MARCHERET, E. and SAVIC, M., "Random walk theory applied to language identification." in *Proceedings of the IEEE International Conference on Acoustics, Signal and Speech Processing*, vol. 2, p. 1119, 1997.
- [29] METZE, F., KEMP, T., SCHAAF, T., SCHULTZ, T., and SOLTAU, H., "Confidence measure based language identification." in *Proceedings of the IEEE International Conference on Acoustics, Signal and Speech Processing*, vol. 3, p. 1827, 2000.
- [30] MUTHUSAMY, Y. K., COLE, R. A., and OSHIKA, B. T., "The OGI multi-language telephone speech corpus." in *Proceedings of the IEEE International Conference on Spoken Language Processing*, vol. 2, pp. 895–898, 1992.
- [31] PAPOULIS, A., *Probability, Random Variables and Stochastic Processes*. Third edition. McGraw-Hill, 1991.
- [32] PEEBLES JR, P. Z., *Probability, Random Variables, and Random Signal Principles*. Third edition. McGraw-Hill, Inc, 1993.
- [33] PROAKIS, J. G. and MANOLAKIS, D. G., *Digital Signal Processing : Principles, Algorithms and Applications*. Third edition. Prentice-Hall, Inc, 1996.
- [34] RABINER, L. R., "A tutorial on Hidden Markov Models and selected applications in speech recognition." in *Proceedings of the IEEE*, vol. 77, 1989.
- [35] REKART, D. M., ZISSMAN, M. A., GLEASON, T. P., and LOSIEWICZ, B. L., "Automatic dialect identification of extemporaneous conversational Latin American Spanish speech." in *Proceedings of the IEEE International Conference on Acoustics, Signal and Speech Processing*, vol. 2, p. 777, 1996.

- [36] REYNOLDS, D. A., QUATIERI, T. F., and DUNN, R. B., "Speaker verification using adapted Gaussian mixture models." *Digital Signal Processing*, Jan 2000, Vol. 10, No. 1, pp. 19–41.
- [37] SCHETZEN, M., *The Volterra and Wiener theories of nonlinear systems*. Wiley and Sons, 1980.
- [38] SCHULTZ, T., ROGINA, I., and WAIBEL, A., "LVCSR-based language identification." in *Proceedings of the IEEE International Conference on Acoustics, Signal and Speech Processing*, vol. 2, p. 781, 1996.
- [39] YAN, Y. and BARNARD, E., "Experiments for an approach to language identification with conversational telephone speech." in *Proceedings of the IEEE International Conference on Acoustics, Signal and Speech Processing*, vol. 2, p. 789, 1996.
- [40] ZISSMAN, M. A., "Automatic language identification using Gaussian mixture and hidden Markov models." in *Proceedings of the IEEE International Conference on Acoustics, Signal and Speech Processing*, vol. 2, p. 399, 1993.
- [41] ZISSMAN, M. A. and SINGER, E., "Automatic language identification of telephone speech messages using phoneme recognition and n-gram modelling." in *Proceedings of the IEEE International Conference on Acoustics, Signal and Speech Processing*, vol. 1, p. 305, 1994.
- [42] ZISSMAN, M. A. and SINGER, E., "Language identification using phoneme recognition and phonotactic language modelling." in *Proceedings of the IEEE International Conference on Acoustics, Signal and Speech Processing*, vol. 5, p. 3503, 1995.
- [43] ZISSMAN, M., "Predicting, diagnosing and improving automatic language identification performance." in *Eurospeech*, 1997.

Appendix A

AST SAE corpus

Table A.1: *SAE corpus speech statistics.*

Utterance durations	SV				
	AE	BE	CE	EE	IE
Total	12h 36m 48s	21h 55m 56s	14h 16m 34s	14h 43m 18s	14h 52m
Shortest	0.42s	0.58s	0.38s	0.97s	0.73s
Longest	1m 37s	9m 59s	2m 24s	5m 58s	3m 24s
Average	6.13s	10.66s	7.02s	7.06s	7.11s
Training	9h 21m 47s	16h 28m 52s	10h 41m 22s	11h 4m 11s	11h 12m 18s
Testing	3h 15m 1s	5h 27m 4s	3h 35m 13s	3h 39m 8s	3h 39m 43s
Cell phone	1h 49m 5s	-	51m 45s	22m 56s	15m 37s
Land line	2h 38m 37s	-	3h 48m 59s	3h 12m 31s	4h 3m 5s
Female	2h 6m 30s	-	1h 54m 58s	2h 24m 21s	2h 48m 37s
Male	2h 21m 12s	-	2h 45m 47s	1h 11m 6s	1h 30m 5s
Unknown	8h 9m 6s	-	9h 35m 50s	10h 58m 51s	10h 33m 18s

Appendix B

Detail experimental results

B.1 Comparison of different preprocessors

Table A.2: *SAE corpus speech statistics following application of PFP preprocessor.*

Utterance durations	SV				
	AE	BE	CE	EE	IE
Total	8h 17m 31s	13h 45m 38s	9h 41m 15s	9h 28m 19s	9h 53m 45s
Shortest	0.17s	0.17s	0.21s	0.54s	0.56s
Longest	1m14s	9m59s	2m7s	3m7s	2m2s
Average	4.03s	6.69s	4.77s	4.54s	4.75s
Training	6h 9m 7s	10h 17m 46s	7h 14m 55s	7h 8m 56s	7h 28m 24s
Testing	2h 8m 24s	3h 27m 52s	2h 26m 21s	2h 19m 24s	2h 25m 21s
Cell phone	1h 14m 34s	-	31m 42s	13m 56s	8m 42s
Land line	1h 47m 47s	-	2h 29m 53s	2h 4m 26s	2h 37m 44s
Female	1h 26m 19s	-	1h 16m 58s	1h 31m 38s	1h 47m 32s
Male	1h 36m 2s	-	1h 44m 37s	46m 44s	58m 54s
Unknown	5h 48m 10s	-	6h 39m 40s	7h 9m 57s	7h 7m 19s

Appendix B

Detail experimental results

B.1 Computation of confidence intervals

For an experiment consisting of n independent Bernoulli trials with k correct classifications, the point estimate x of the experiment expected value p is given by [31]:

$$x = k/n. \tag{B.1}$$

The interval estimate of p is given by:

$$p = \frac{z_u^2/n + 2x \pm \sqrt{(z_u^2/n + 2x)^2 - 4x^2(1 + z_u^2/n)}}{2(1 + z_u^2/n)}, \tag{B.2}$$

with z_u the u percentile of the standard normal density. This equation yields $p_1 < p < p_2$, the interval estimate of p . Use $z_u = 1.645$ for a 95% confidence interval [31].

Table B.2: 95% confidence intervals

Test segment length [s]	AE	BE	CE	DE	FE	Agreement
2	53.63	45.35	37.25	33.47	30.39	25.31, 37.25, 53.63
4	50.33	45.6	38.87	33.93	31.9	25.1, 37.45, 50.33
10	43.88	45.61	38.57	33.81	31.58	25.34, 37.46, 50.33
30	40.35	45.3	38.16	33.78	31.52	25.34, 37.46, 50.33
60	39.84	45.38	38.63	33.25	31.71	25.35, 37.78, 50.33
120	37.5	44.68	38.49	33.72	31.83	25.95, 37.99, 50.33
300	32.6	35.59	30.69	33.33	31.14	25.29, 37.16, 50.33

B.2 Tables and graphs for experiments

Table B.1: *CARP configuration RER [%].*

Test segment length [s]	AE	BE	CE	EE	IE	Average
2	62,17	44,49	88,28	94,21	66,3	68,74 [68,33 : 69,14]
4	57,88	45,74	86,33	94,81	62,07	67,34 [66,75 : 67,92]
10	50,9	48,01	82,71	92,14	56,94	64,69 [63,75 : 65,62]
30	41,65	50,15	75,23	83,94	45,1	58,65 [56,97 : 60,31]
60	37,63	52,29	62,62	72,48	38,36	52,9 [50,5 : 55,29]
120	38,14	54,6	51,4	58,72	26,61	46,84 [43,47 : 50,24]
300	28,95	50,77	38,1	44,19	9,3	35,93 [30,93 : 41,26]

Table B.2: *PFP configuration RER [%].*

Test segment length [s]	AE	BE	CE	EE	IE	Average
2	53,83	45,25	87,85	95,47	68,38	68,31 [67,8 : 68,81]
4	50,23	45,6	83,87	93,91	64,9	66,1 [65,37 : 66,82]
10	43,88	45,62	80,57	89,81	57,88	62,34 [61,16 : 63,51]
30	40,23	45,3	69,76	78,78	41,52	54,48 [52,38 : 56,57]
60	39,84	46,38	58,62	76,26	34,72	50,85 [47,88 : 53,82]
120	37,5	44,66	38,89	50,72	20,83	38,95 [34,92 : 43,13]
300	12,0	36,59	20,69	33,33	7,14	23,33 [18,16 : 29,46]

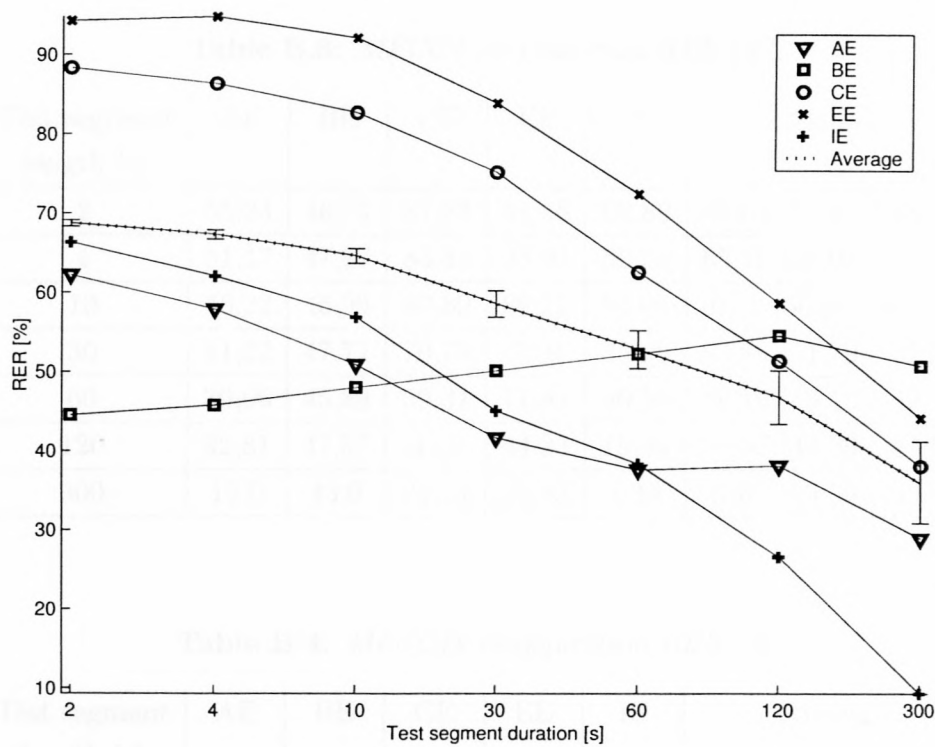


Figure B.1: CARP configuration RER [%].

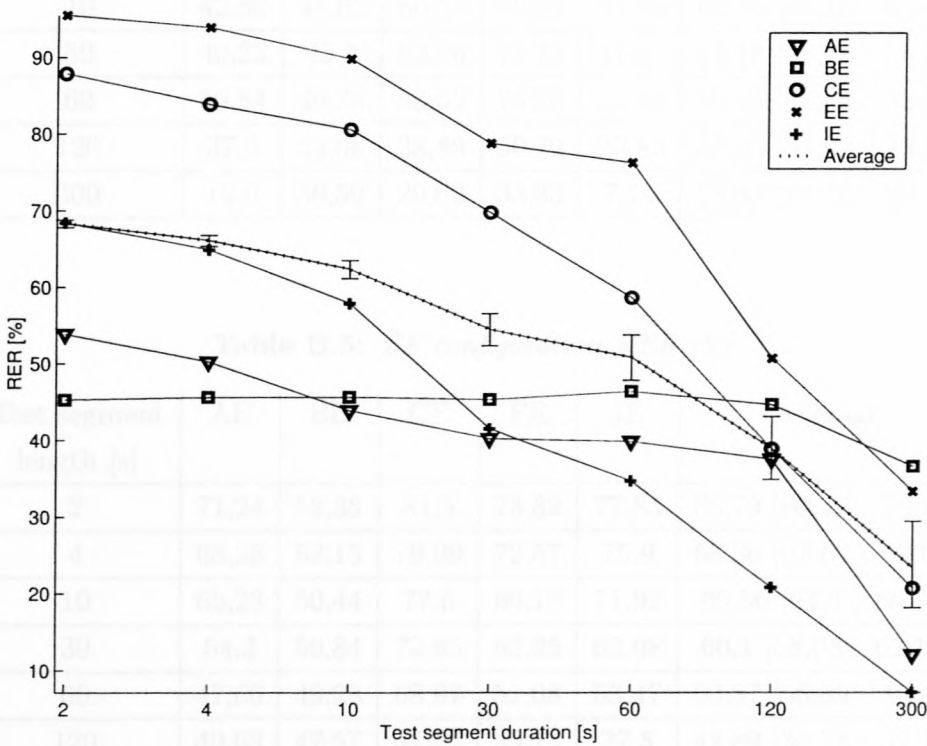


Figure B.2: PFP configuration RER [%].

Table B.3: *MFCC9 configuration RER [%].*

Test segment length [s]	AE	BE	CE	EE	IE	Average
2	55,34	46,73	87,23	94,58	62,89	67,64 [67,13 : 68,15]
4	51,17	47,17	83,83	93,91	60,03	65,75 [65,02 : 66,48]
10	46,22	46,99	80,69	89,21	53,05	62,1 [60,92 : 63,27]
30	41,02	47,71	70,79	75,9	35,99	53,89 [51,79 : 55,98]
60	39,06	45,89	59,31	71,94	30,56	49,15 [46,18 : 52,12]
120	32,81	47,57	37,5	44,93	19,44	37,37 [33,39 : 41,53]
300	12,0	43,9	24,14	25,93	7,14	24,67 [19,36 : 30,87]

Table B.4: *MFCC18 configuration RER [%].*

Test segment length [s]	AE	BE	CE	EE	IE	Average
2	53,83	45,25	87,85	95,47	68,38	68,31 [67,8 : 68,81]
4	50,23	45,6	83,87	93,91	64,9	66,1 [65,37 : 66,82]
10	43,88	45,62	80,57	89,81	57,88	62,34 [61,16 : 63,51]
30	40,23	45,3	69,76	78,78	41,52	54,48 [52,38 : 56,57]
60	39,84	46,38	58,62	76,26	34,72	50,85 [47,88 : 53,82]
120	37,5	44,66	38,89	50,72	20,83	38,95 [34,92 : 43,13]
300	12,0	36,59	20,69	33,33	7,14	23,33 [18,16 : 29,46]

Table B.5: *FF configuration RER [%].*

Test segment length [s]	AE	BE	CE	EE	IE	Average
2	71,24	52,35	81,5	73,82	77,83	69,79 [69,29 : 70,29]
4	68,25	52,15	79,99	72,57	75,9	68,36 [67,64 : 69,07]
10	65,23	50,44	77,6	69,18	71,92	65,56 [64,4 : 66,71]
30	54,3	50,84	72,85	62,95	62,98	60,1 [58,03 : 62,15]
60	47,66	49,28	68,97	51,08	53,47	53,87 [50,89 : 56,82]
120	40,63	47,57	47,22	39,13	37,5	42,89 [38,78 : 47,11]
300	28,0	48,78	34,48	25,93	17,86	32,67 [26,72 : 39,22]

Table B.6: *MFCC9_FF configuration RER [%].*

Test segment length [s]	AE	BE	CE	EE	IE	Average
2	55,47	47,81	84,56	92,64	63,09	67,13 [66,62 : 67,64]
4	52,21	47,91	82,0	90,79	60,03	65,21 [64,48 : 65,94]
10	46,22	47,79	79,43	85,73	53,62	61,56 [60,37 : 62,73]
30	41,8	47,71	66,67	73,02	38,75	53,24 [51,13 : 55,33]
60	38,28	45,89	57,93	64,03	30,56	47,31 [44,35 : 50,29]
120	31,25	48,54	40,28	46,38	22,22	38,68 [34,67 : 42,86]
300	12,0	43,9	17,24	18,52	3,57	21,33 [16,36 : 27,32]

Table B.7: *AH configuration RER [%].*

Test segment length [s]	AE	BE	CE	EE	IE	Average
2	52,45	44,56	91,71	94,61	66,5	68,11 [67,61 : 68,62]
4	48,78	44,47	88,99	93,09	63,48	66,11 [65,38 : 66,83]
10	42,32	45,54	86,74	88,85	57,42	62,97 [61,79 : 64,14]
30	37,5	45,54	74,91	77,34	43,6	55,2 [53,1 : 57,28]
60	38,28	44,44	65,52	74,1	40,28	52,03 [49,05 : 54,99]
120	31,25	48,54	44,44	55,07	26,39	41,84 [37,75 : 46,05]
300	16,0	39,02	31,03	25,93	10,71	26,0 [20,57 : 32,28]

Table B.8: *PLP configuration RER [%].*

Test segment length [s]	AE	BE	CE	EE	IE	Average
2	55,54	48,11	83,26	81,06	64,31	65,11 [64,59 : 65,62]
4	51,8	46,88	81,09	77,36	60,99	62,43 [61,69 : 63,17]
10	47,4	44,98	78,63	71,7	53,51	58,27 [57,06 : 59,46]
30	42,58	43,13	67,7	58,63	37,02	49,38 [47,28 : 51,48]
60	35,94	37,68	51,03	48,2	27,78	39,97 [37,1 : 42,92]
120	32,81	37,86	38,89	36,23	11,11	31,84 [28,05 : 35,89]
300	12,0	26,83	6,9	18,52	0,0	14,0 [9,98 : 19,3]

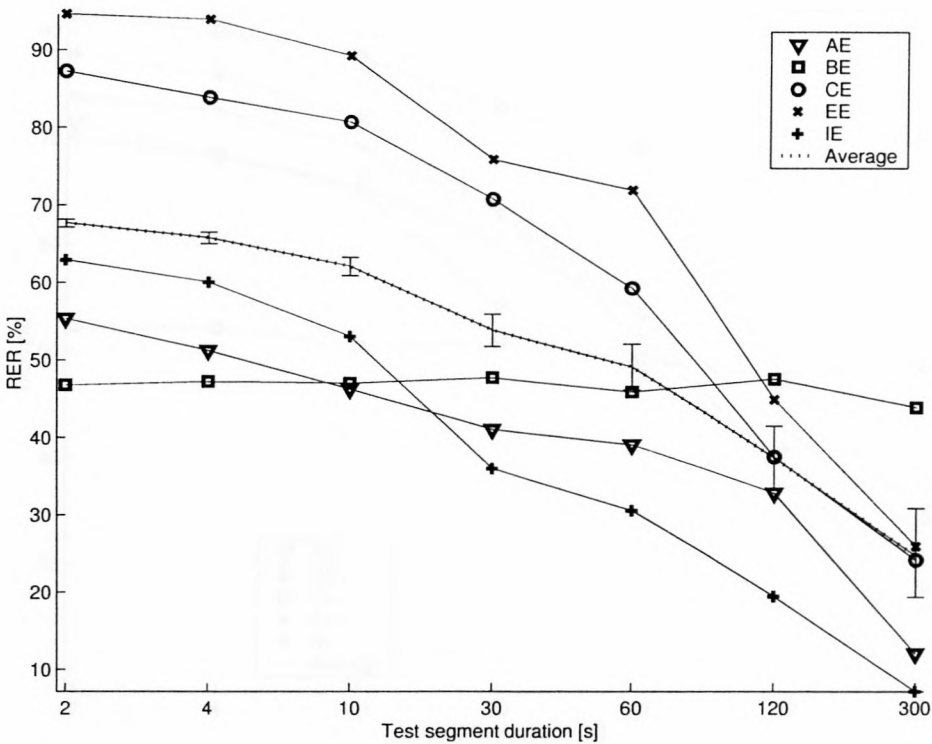


Figure B.3: MFCC9 configuration RER [%].

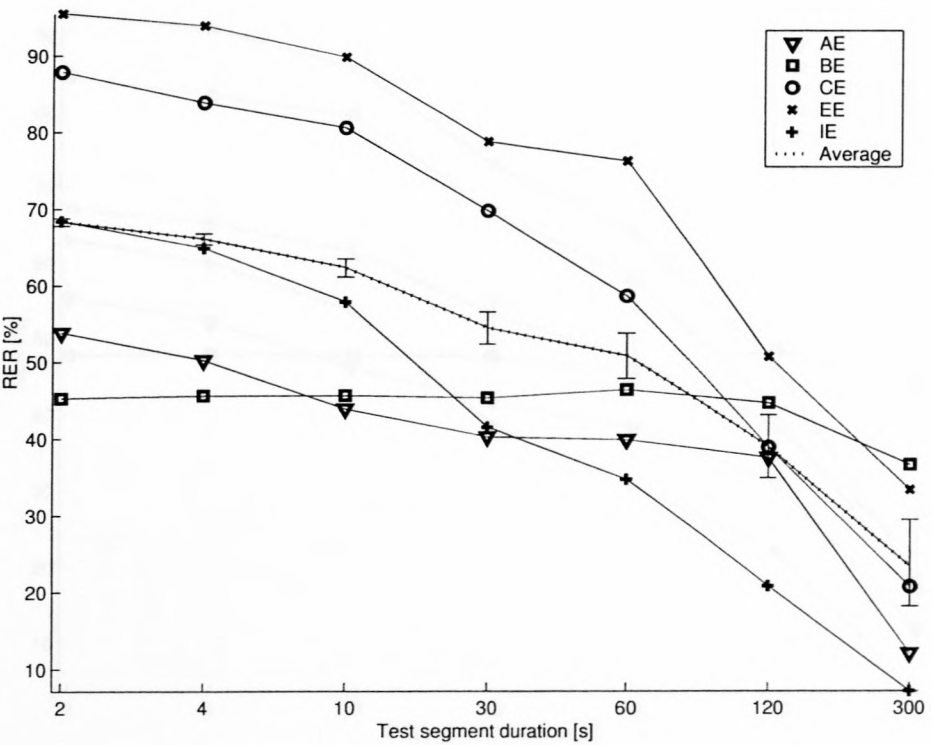


Figure B.4: MFCC18 configuration RER [%].

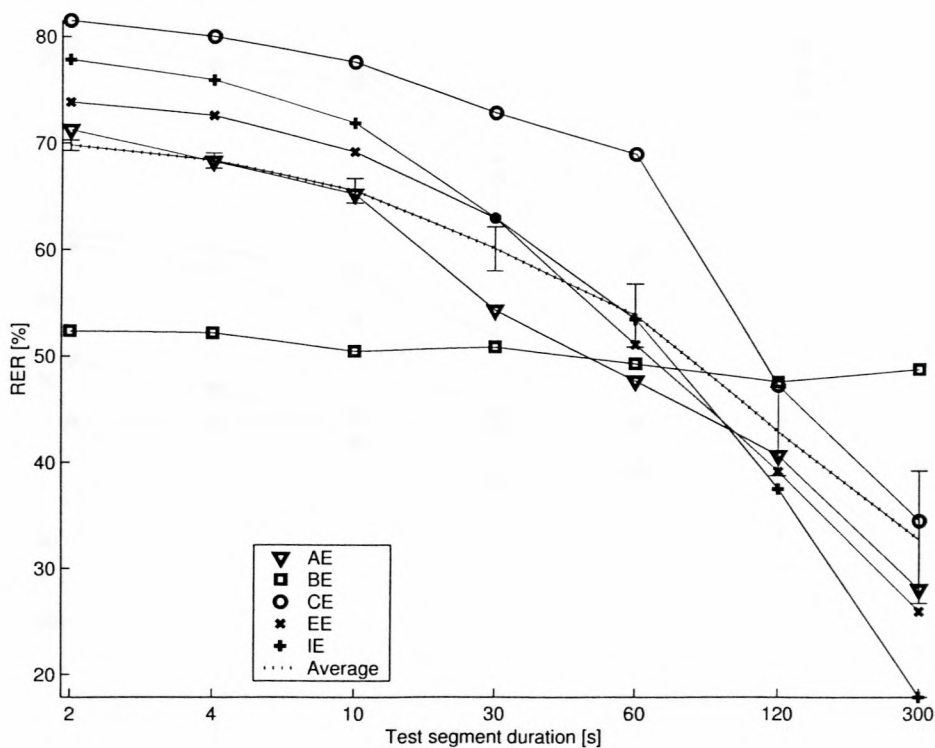


Figure B.5: *FF* configuration RER [%].

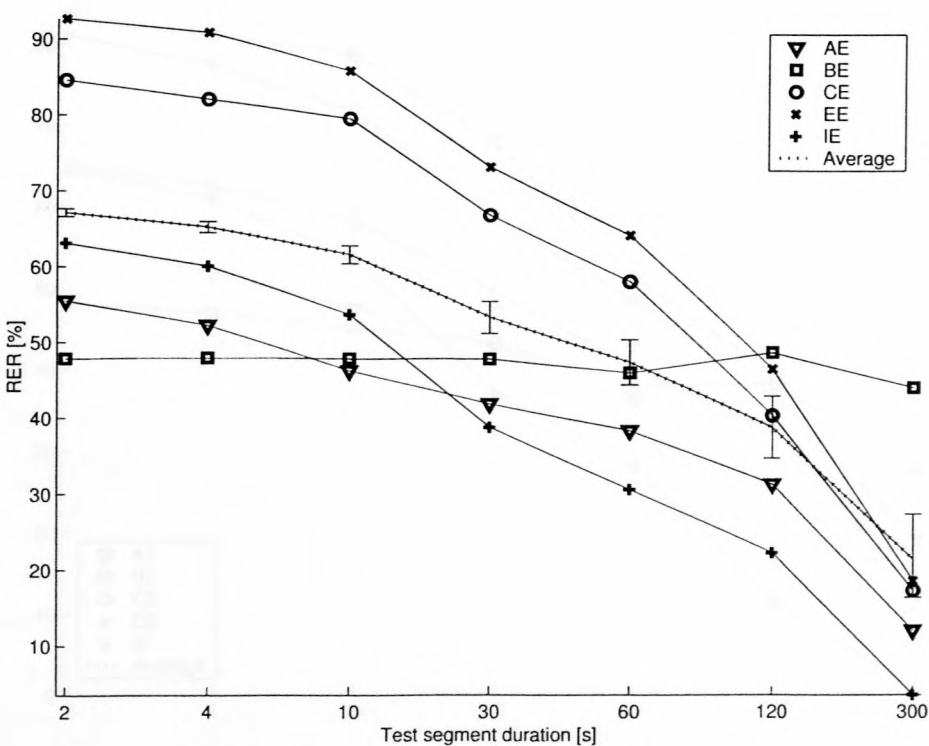


Figure B.6: *MFCC9_FF* configuration RER [%].

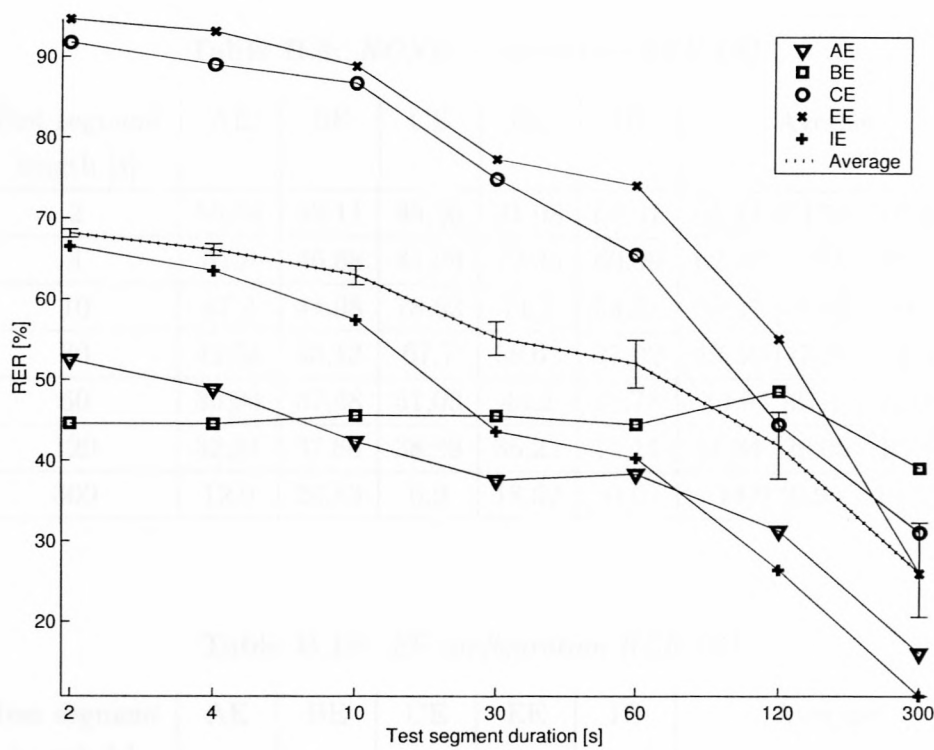


Figure B.7: AH configuration RER [%].

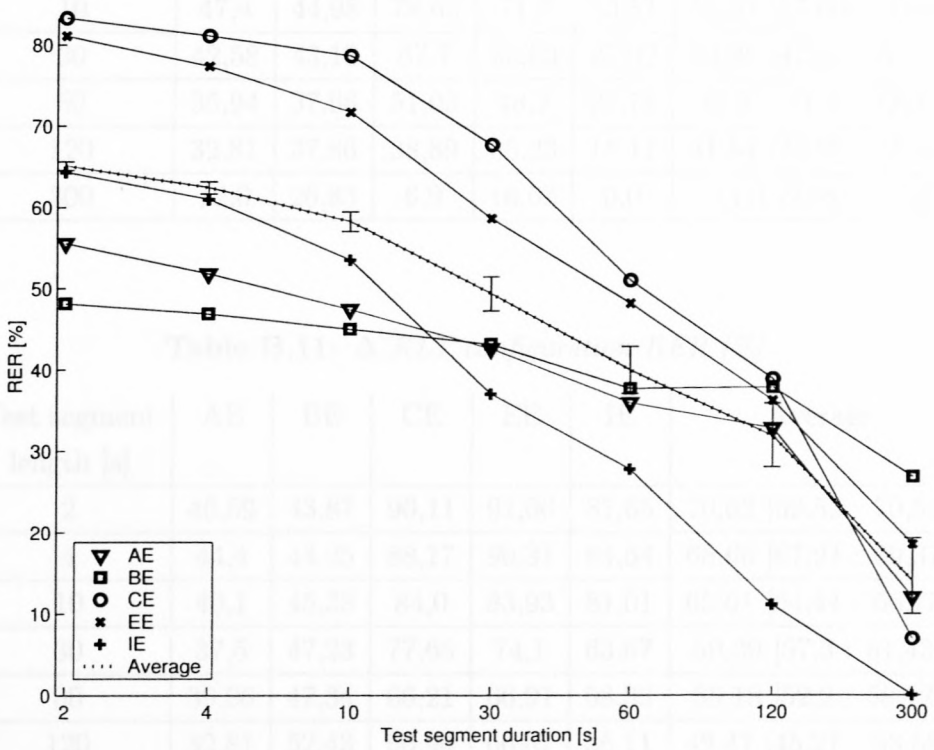


Figure B.8: PLP configuration RER [%].

Table B.9: *NONE configuration RER [%].*

Test segment length [s]	AE	BE	CE	EE	IE	Average
2	55,54	48,11	83,26	81,08	64,31	65,11 [64,59 : 65,63]
4	51,8	46,88	81,09	77,36	60,99	62,43 [61,69 : 63,17]
10	47,4	44,98	78,63	71,7	53,51	58,27 [57,06 : 59,46]
30	42,58	43,13	67,7	58,63	37,02	49,38 [47,28 : 51,48]
60	35,94	37,68	51,03	48,2	27,78	39,97 [37,1 : 42,92]
120	32,81	37,86	38,89	36,23	11,11	31,84 [28,05 : 35,89]
300	12,0	26,83	6,9	18,52	0,0	14,0 [9,98 : 19,3]

Table B.10: *FS configuration RER [%].*

Test segment length [s]	AE	BE	CE	EE	IE	Average
2	55,54	48,11	83,26	81,06	64,31	65,11 [64,59 : 65,62]
4	51,8	46,88	81,09	77,36	60,99	62,43 [61,69 : 63,17]
10	47,4	44,98	78,63	71,7	53,51	58,27 [57,06 : 59,46]
30	42,58	43,13	67,7	58,63	37,02	49,38 [47,28 : 51,48]
60	35,94	37,68	51,03	48,2	27,78	39,97 [37,1 : 42,92]
120	32,81	37,86	38,89	36,23	11,11	31,84 [28,05 : 35,89]
300	12,0	26,83	6,9	18,52	0,0	14,0 [9,98 : 19,3]

Table B.11: Δ_{KLT} configuration RER [%].

Test segment length [s]	AE	BE	CE	EE	IE	Average
2	46,69	43,87	90,11	91,06	87,65	70,02 [69,52 : 70,52]
4	44,4	44,25	88,17	90,31	84,64	68,66 [67,94 : 69,37]
10	40,1	45,38	84,0	83,93	81,01	65,61 [64,44 : 66,75]
30	37,5	47,23	77,66	74,1	63,67	59,39 [57,3 : 61,43]
60	39,06	47,34	66,21	66,91	58,33	55,18 [52,2 : 58,12]
120	32,81	52,43	56,94	66,67	36,11	49,47 [45,27 : 53,68]
300	16,0	53,66	34,48	40,74	21,43	35,33 [29,22 : 41,96]

Table B.12: $\Delta\Delta_KLT$ configuration RER [%].

Test segment length [s]	AE	BE	CE	EE	IE	Average
2	47,01	40,71	89,72	93,02	90,48	70,03 [69,53 : 70,53]
4	44,14	41,36	87,71	91,8	87,9	68,64 [67,92 : 69,34]
10	40,76	43,21	85,14	88,25	81,93	66,3 [65,15 : 67,44]
30	39,84	45,3	78,35	78,06	73,7	62,0 [59,94 : 64,02]
60	41,41	45,41	66,9	69,06	61,11	56,09 [53,12 : 59,02]
120	31,25	51,46	61,11	65,22	40,28	50,26 [46,06 : 54,47]
300	28,0	53,66	48,28	44,44	14,29	39,33 [33,02 : 46,03]

Table B.13: KLT configuration RER [%].

Test segment length [s]	AE	BE	CE	EE	IE	Average
2	53,75	45,27	79,29	82,67	67,39	64,16 [63,63 : 64,67]
4	49,92	44,02	76,11	79,62	63,89	61,35 [60,6 : 62,1]
10	46,09	39,68	72,46	72,06	54,43	55,67 [54,47 : 56,88]
30	41,41	36,39	57,39	61,15	39,45	46,3 [44,22 : 48,41]
60	33,59	28,99	41,38	47,48	28,47	35,39 [32,6 : 38,28]
120	29,69	26,21	30,56	28,99	16,67	26,32 [22,78 : 30,19]
300	12,0	12,2	13,79	14,81	3,57	11,33 [7,74 : 16,29]

Table B.14: FFE_KLT configuration RER [%].

Test segment length [s]	AE	BE	CE	EE	IE	Average
2	52,16	47,08	87,57	82,49	73,93	67,16 [66,65 : 67,67]
4	48,72	47,5	86,38	80,25	69,47	65,22 [64,48 : 65,95]
10	46,2	45,45	82,68	74,7	63,51	61,4 [60,21 : 62,58]
30	39,76	44,07	73,1	63,41	48,26	53,19 [51,08 : 55,29]
60	37,8	43,69	64,14	54,35	34,03	46,71 [43,75 : 49,69]
120	39,68	37,86	48,61	37,68	27,78	38,26 [34,25 : 42,43]
300	16,0	31,71	34,48	22,22	10,71	24,0 [18,76 : 30,16]

Table B.15: *FCE_KLT* configuration *RER* [%].

Test segment length [s]	AE	BE	CE	EE	IE	Average
2	57,6	42,43	79,15	87,6	74,18	66,18 [65,67 : 66,7]
4	53,36	44,31	77,16	83,07	70,24	64,04 [63,3 : 64,77]
10	48,7	43,78	72,57	74,7	63,64	59,46 [58,27 : 60,65]
30	39,84	43,37	62,89	59,35	47,75	50,23 [48,13 : 52,33]
60	35,94	36,71	51,72	44,6	38,89	41,28 [38,39 : 44,24]
120	37,5	41,75	40,28	37,68	23,61	36,58 [32,62 : 40,73]
300	16,0	31,71	24,14	22,22	7,14	21,33 [16,36 : 27,32]

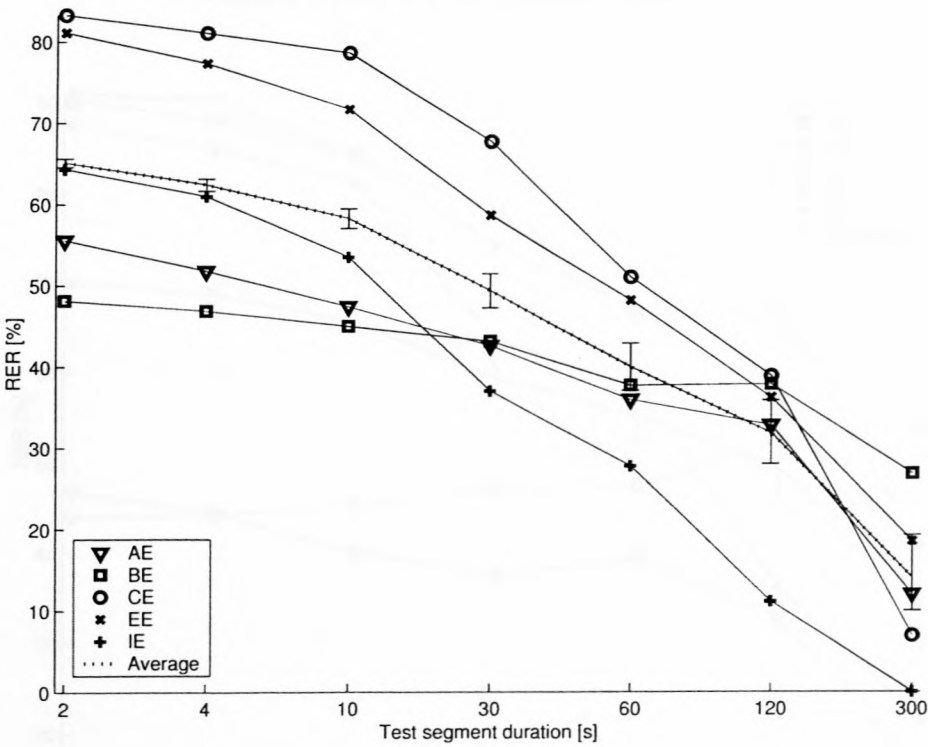


Figure B.9: *NONE* configuration *RER* [%].

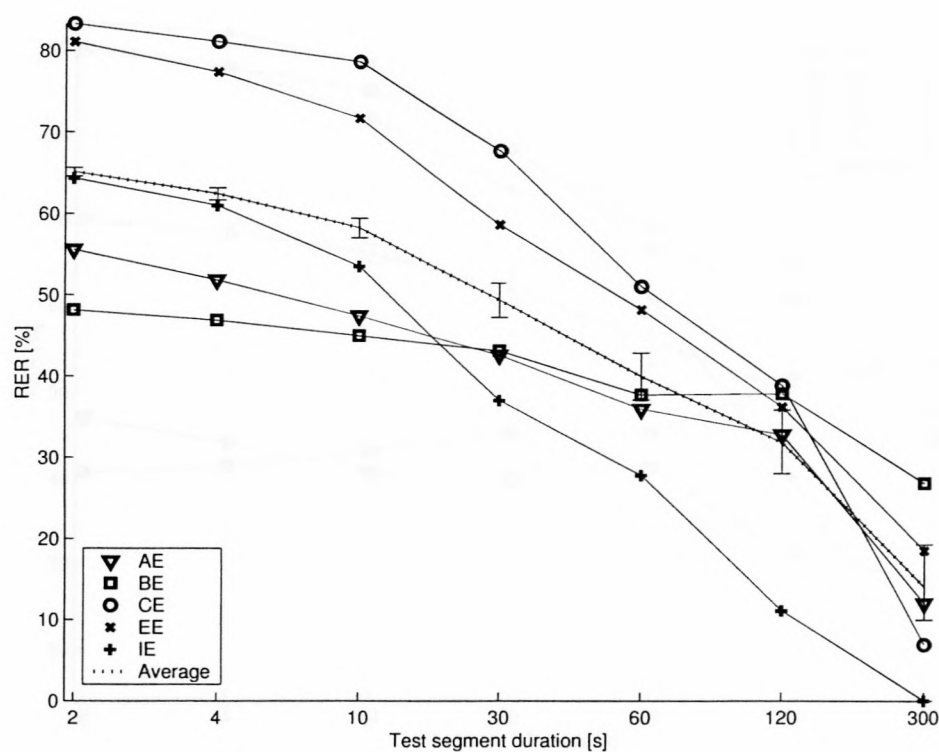


Figure B.10: *FS configuration RER [%].*

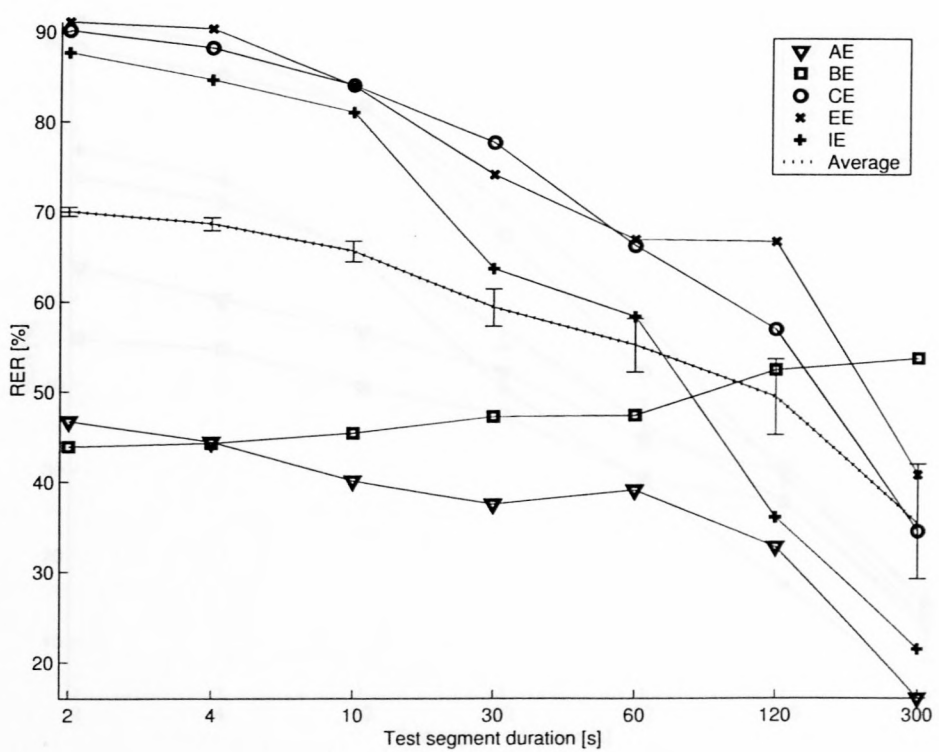


Figure B.11: Δ_{KLT} configuration RER [%].

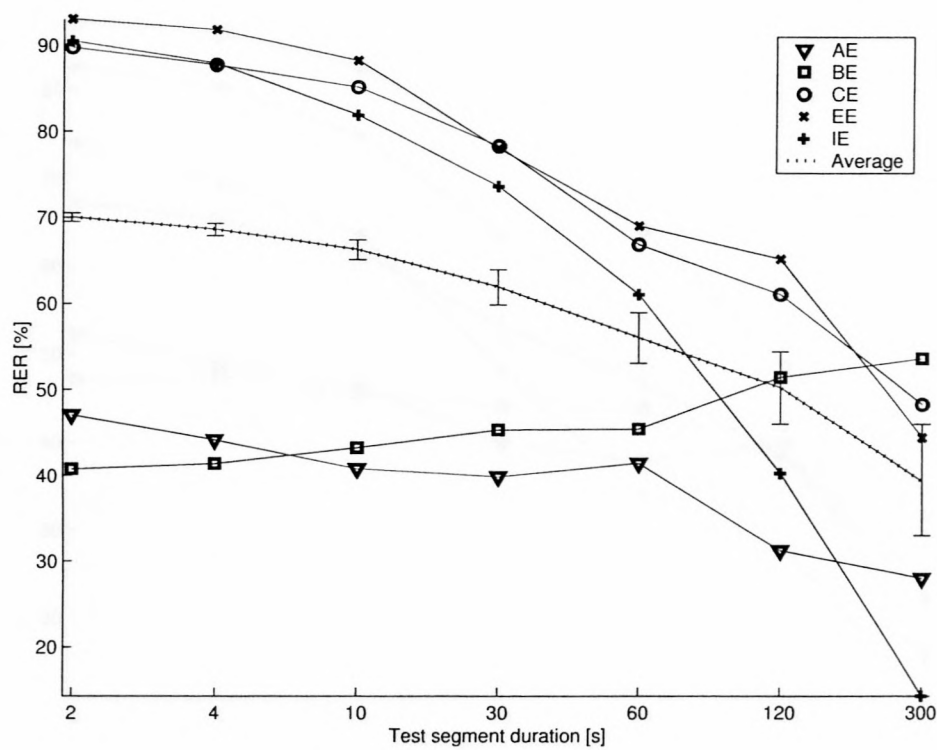


Figure B.12: $\Delta\Delta_{KLT}$ configuration RER [%].

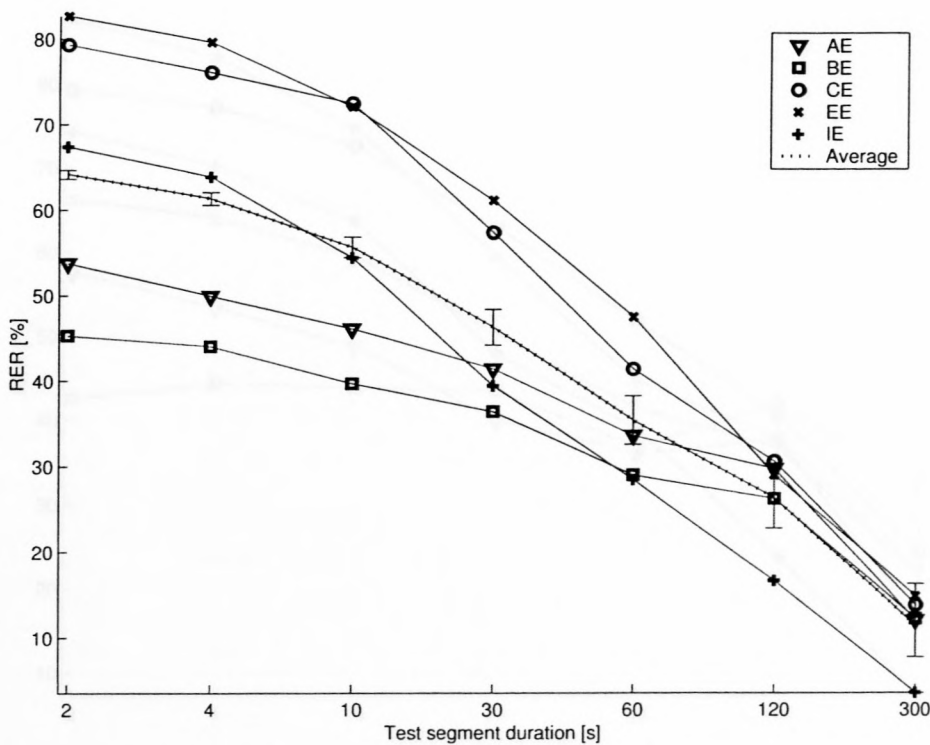


Figure B.13: KLT configuration RER [%].

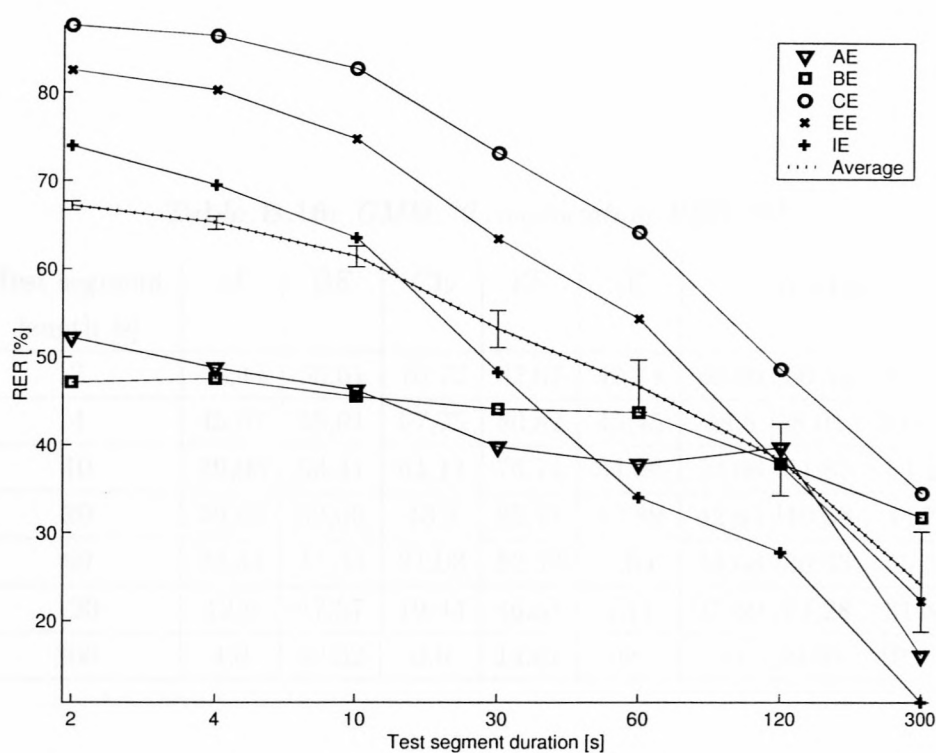


Figure B.14: *FFE_KLT* configuration *RER* [%].

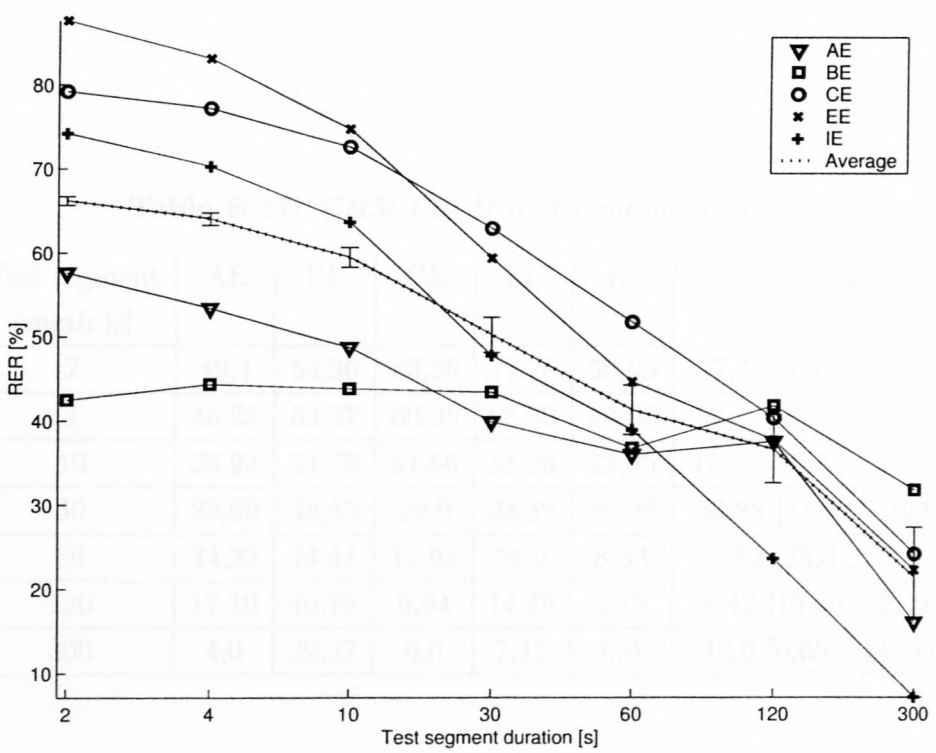


Figure B.15: *FCE_KLT* configuration *RER* [%].

Table B.16: *GMM_70 configuration RER [%].*

Test segment length [s]	AE	BE	CE	EE	IE	Average
2	48,44	55,61	70,72	82,07	49,78	60,99 [60,46 : 61,52]
4	45,97	55,01	67,75	80,82	45,45	58,8 [58,04 : 59,55]
10	39,06	53,41	61,14	76,74	34,06	53,06 [51,85 : 54,27]
30	29,69	52,05	43,3	65,47	17,99	42,64 [40,58 : 44,73]
60	23,44	47,34	31,03	52,52	7,64	33,68 [30,93 : 36,55]
120	12,5	47,57	19,44	46,38	4,17	27,89 [24,28 : 31,83]
300	4,0	39,02	0,0	14,81	0,0	14,0 [9,98 : 19,3]

Table B.17: *GMM_CSVM configuration RER [%].*

Test segment length [s]	AE	BE	CE	EE	IE	Average
2	49,4	54,36	63,96	71,76	50,03	57,7 [57,17 : 58,24]
4	46,23	53,37	60,35	68,59	45,08	54,7 [53,93 : 55,46]
10	38,93	51,73	51,66	58,99	33,95	47,53 [46,32 : 48,74]
30	29,69	48,43	29,9	38,49	16,26	33,88 [31,92 : 35,9]
60	24,22	44,44	17,93	25,9	8,33	25,82 [23,3 : 28,51]
120	17,19	40,78	6,94	14,49	2,78	18,42 [15,38 : 21,91]
300	4,0	29,27	0,0	7,41	0,0	10,0 [6,65 : 14,76]

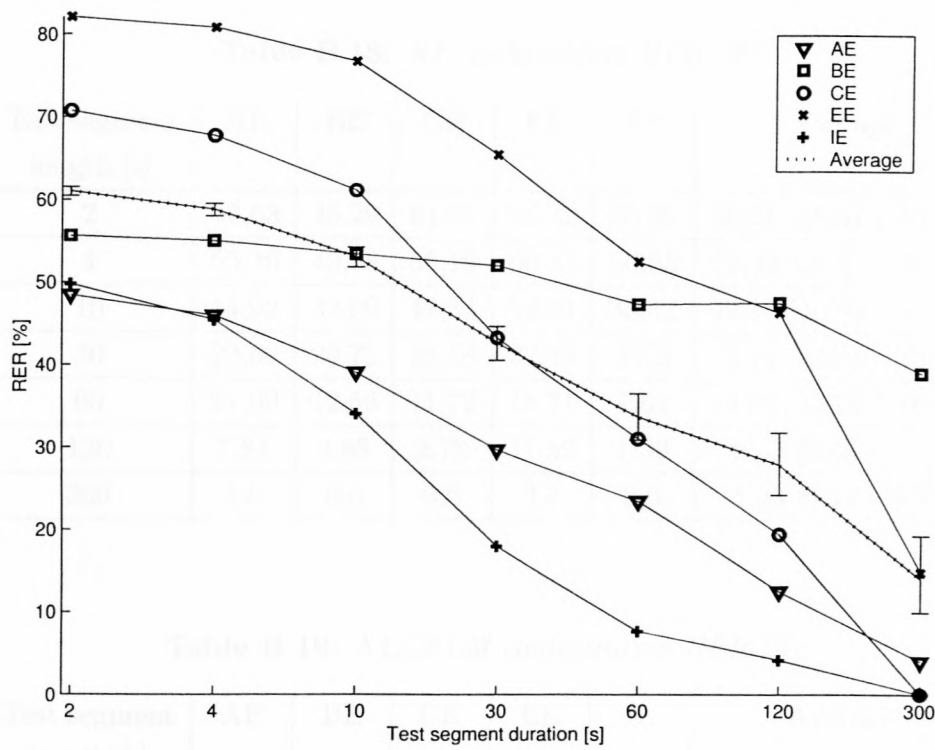


Figure B.16: *GMM_70* configuration RER [%].

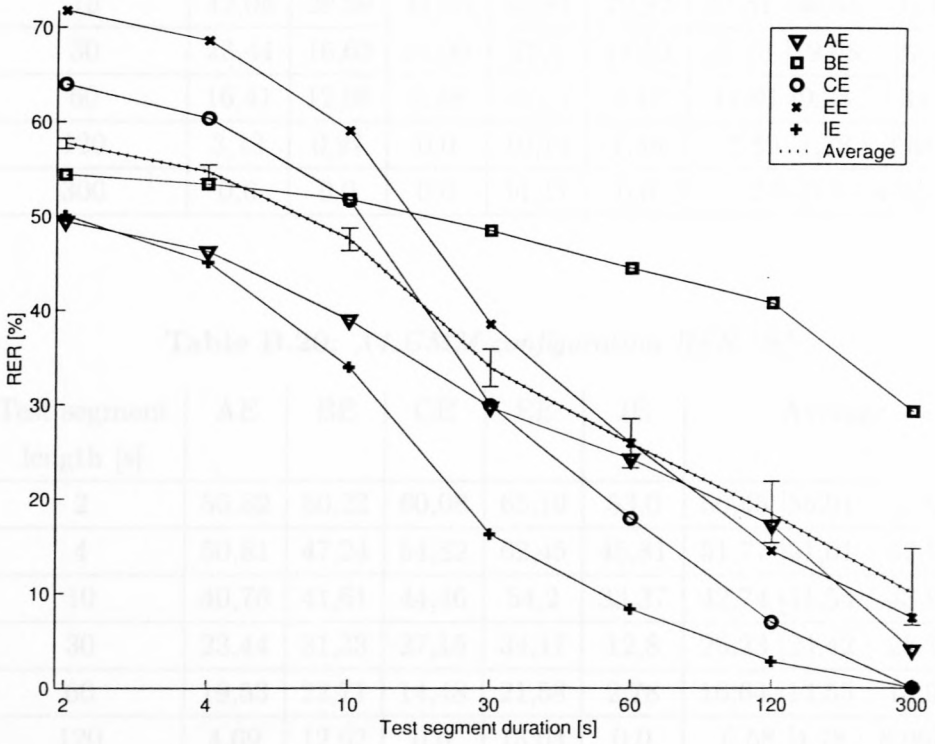


Figure B.17: *GMM_CSVM* configuration RER [%].

Table B.18: *X1 configuration RER [%].*

Test segment length [s]	AE	BE	CE	EE	IE	Average
2	57,63	45,29	61,04	66,32	56,06	56,21 [55,67 : 56,75]
4	53,46	40,58	56,19	60,82	50,55	51,28 [50,51 : 52,04]
10	44,92	32,69	47,54	52,04	38,32	42,15 [40,95 : 43,35]
30	29,69	20,72	28,18	32,37	17,3	25,11 [23,34 : 26,98]
60	21,09	12,56	11,72	18,71	7,64	14,02 [12,08 : 16,22]
120	7,81	4,85	2,78	11,59	1,39	5,53 [3,89 : 7,79]
300	4,0	0,0	0,0	3,7	0,0	1,33 [0,44 : 3,95]

Table B.19: *X1_CSV configuration RER [%].*

Test segment length [s]	AE	BE	CE	EE	IE	Average
2	57,37	40,39	57,41	65,91	49,32	52,8 [52,26 : 53,34]
4	51,59	36,25	51,9	61,15	41,63	47,34 [46,57 : 48,11]
10	42,06	28,59	41,03	50,84	29,92	37,51 [36,34 : 38,69]
30	23,44	16,63	21,99	29,5	14,53	20,73 [19,08 : 22,49]
60	16,41	12,08	8,28	15,11	3,47	11,01 [9,28 : 13,01]
120	3,13	0,97	0,0	10,14	1,39	2,89 [1,78 : 4,68]
300	0,0	0,0	0,0	11,11	0,0	2,0 [0,8 : 4,9]

Table B.20: *X1_GMM configuration RER [%].*

Test segment length [s]	AE	BE	CE	EE	IE	Average
2	56,82	50,23	60,08	65,19	53,0	56,45 [55,91 : 56,99]
4	50,81	47,24	54,82	62,45	45,81	51,77 [51,01 : 52,54]
10	40,76	41,61	44,46	54,2	33,37	42,74 [41,54 : 43,94]
30	23,44	31,33	27,15	34,17	12,8	26,23 [24,42 : 28,12]
60	19,53	22,71	14,48	21,58	2,78	16,64 [14,55 : 18,98]
120	4,69	12,62	0,0	13,04	0,0	6,58 [4,78 : 8,99]
300	0,0	0,0	0,0	11,11	0,0	2,0 [0,8 : 4,9]

Table B.21: *X1_GMM_CSVM configuration RER [%].*

Test segment length [s]	AE	BE	CE	EE	IE	Average
2	54,87	40,88	57,98	63,94	48,72	52,16 [51,61 : 52,7]
4	50,13	36,09	53,22	58,18	41,44	46,73 [45,96 : 47,5]
10	38,15	28,59	42,97	48,32	30,03	36,79 [35,63 : 37,97]
30	19,53	15,18	21,65	26,26	14,19	18,97 [17,37 : 20,67]
60	8,59	8,21	9,66	17,99	4,17	9,57 [7,96 : 11,46]
120	1,56	0,97	1,39	7,25	0,0	2,11 [1,19 : 3,7]
300	0,0	0,0	0,0	7,41	0,0	1,33 [0,44 : 3,95]

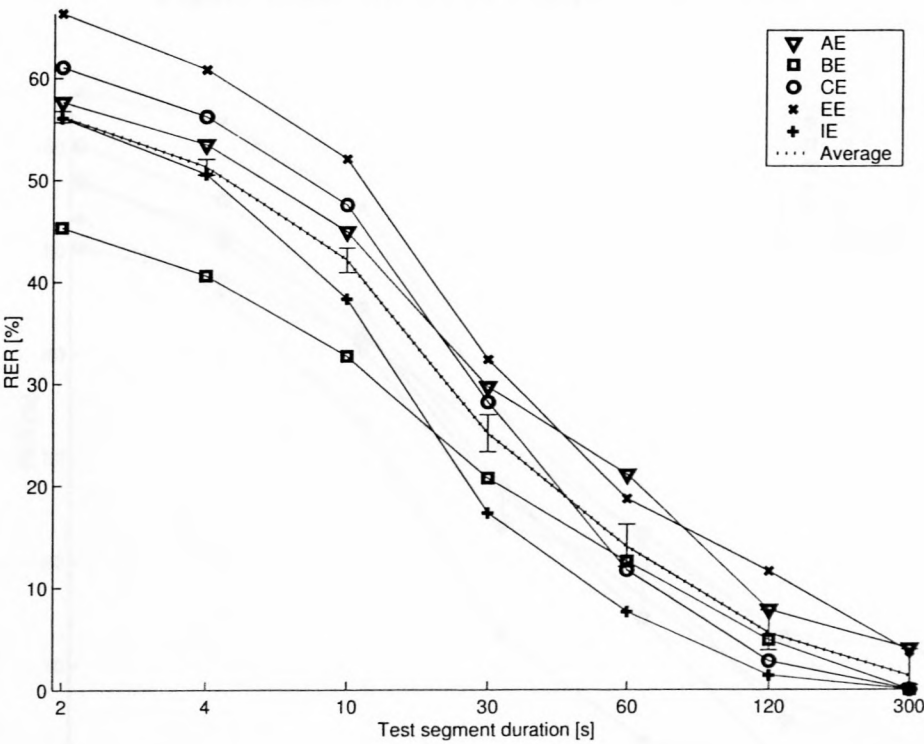


Figure B.18: *X1 configuration RER [%].*

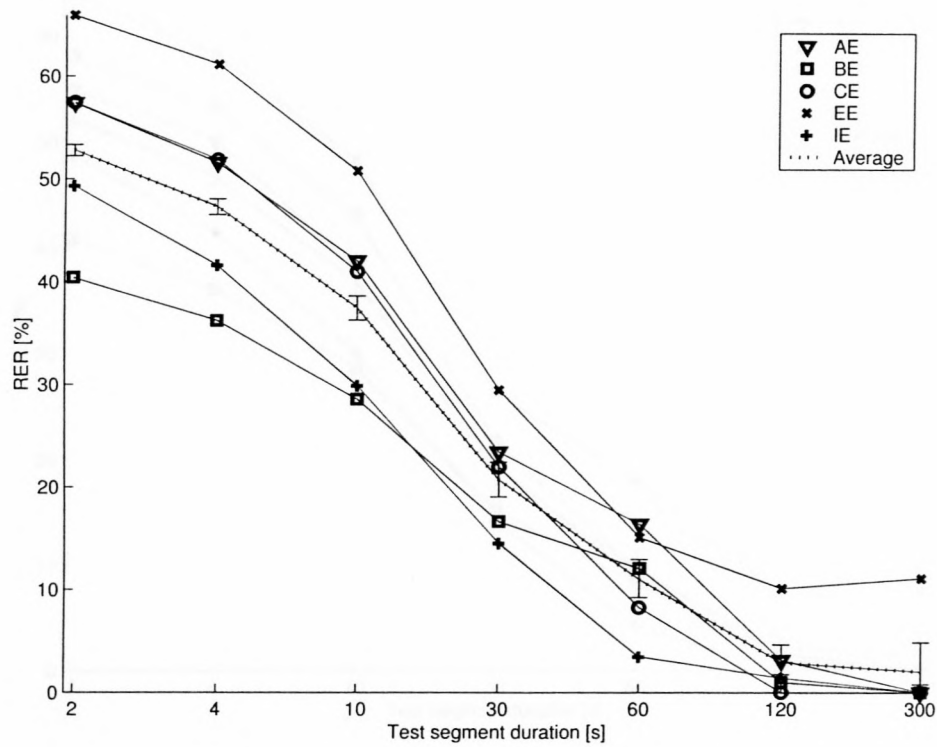


Figure B.19: *X1_CSVM* configuration RER [%].

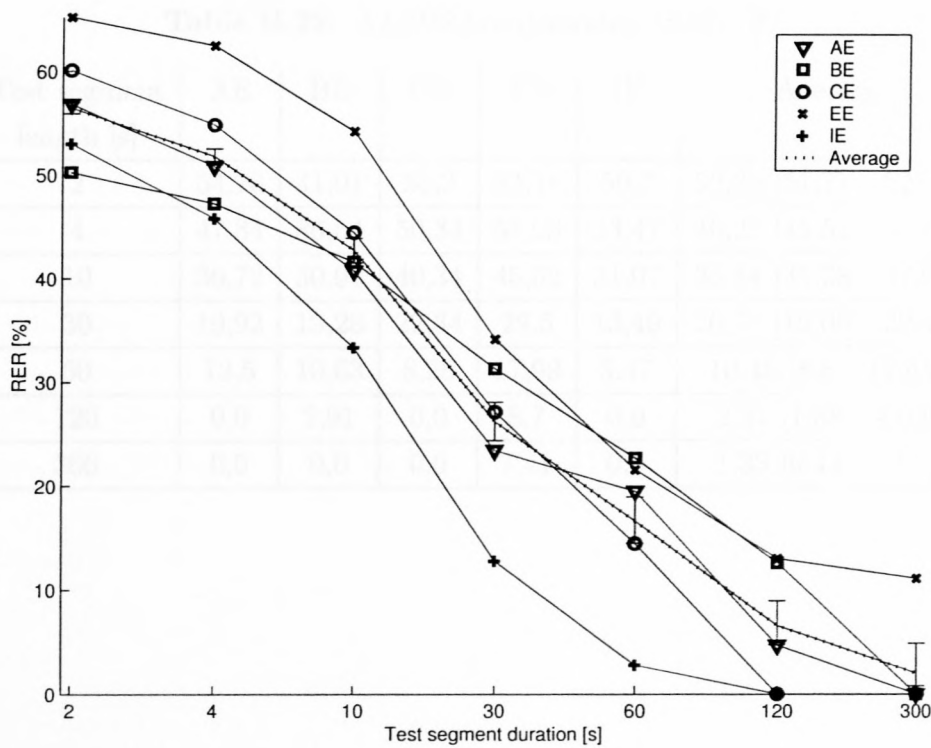


Figure B.20: *X1_GMM* configuration RER [%].

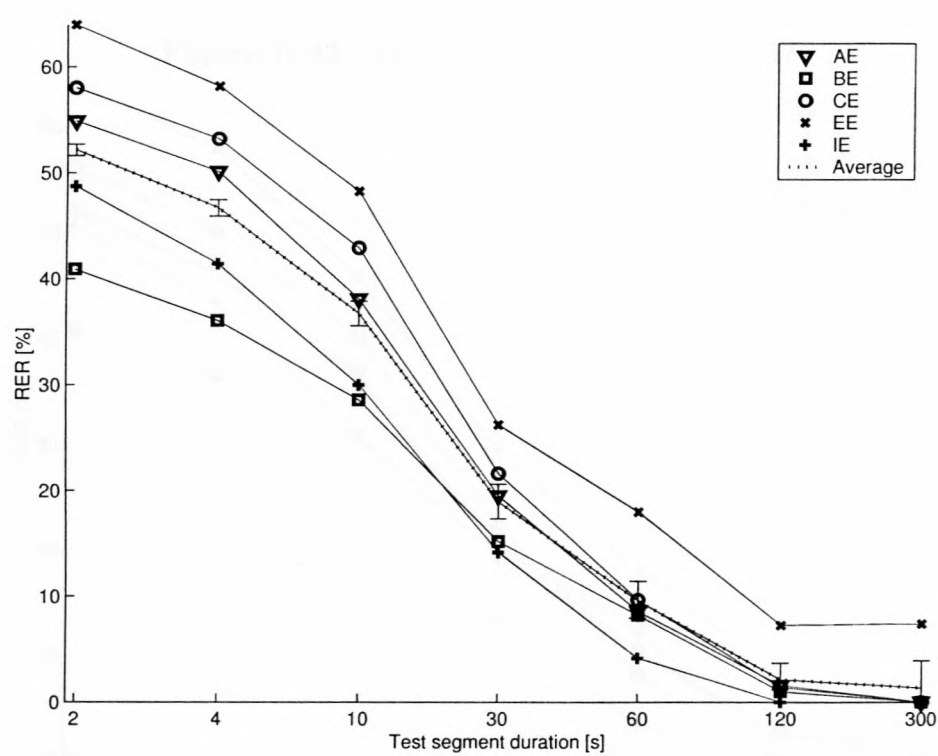


Figure B.21: *X1_GMM_CSV* configuration *RER* [%].

Table B.22: *X1_HVQ* configuration *RER* [%].

Test segment length [s]	AE	BE	CE	EE	IE	Average
2	54,76	41,01	56,2	64,18	50,7	52,25 [51,71 : 52,79]
4	47,84	36,54	50,34	58,08	43,47	46,29 [45,52 : 47,05]
10	36,72	30,92	40,34	45,92	31,07	36,44 [35,28 : 37,62]
30	19,92	19,28	22,34	29,5	13,49	20,73 [19,08 : 22,49]
60	12,5	10,63	8,28	17,99	3,47	10,48 [8,8 : 12,45]
120	0,0	2,91	0,0	8,7	0,0	2,37 [1,38 : 4,03]
300	0,0	0,0	0,0	7,41	0,0	1,33 [0,44 : 3,95]

Figure B.22: *X1_HVQ configuration RER [%]*.

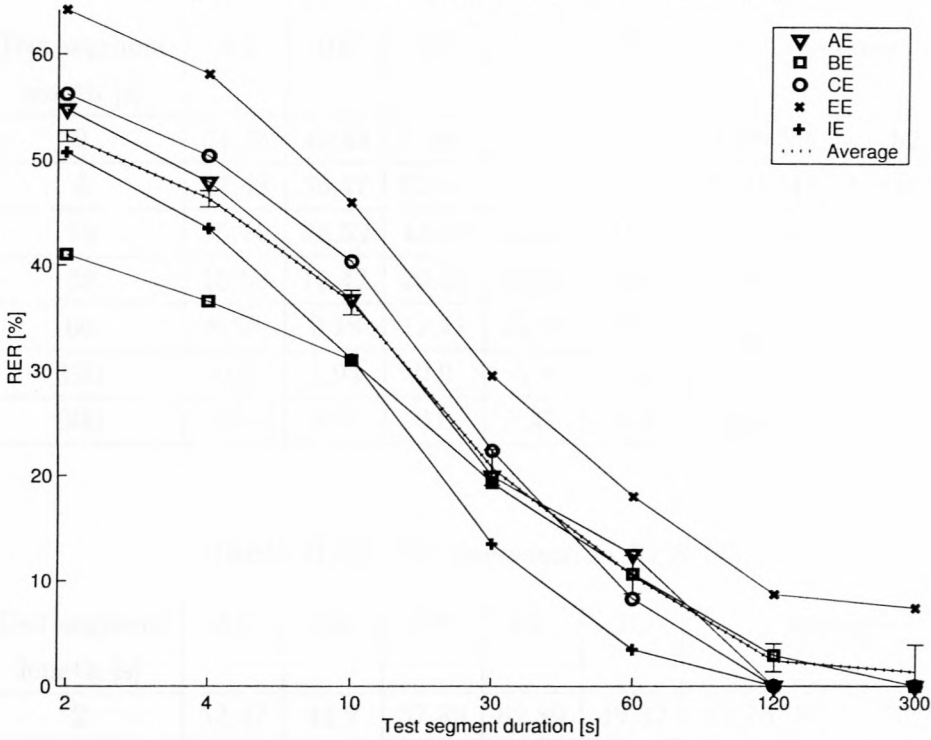


Table B.23: *X2 configuration RER [%]*.

Test segment length [s]	AE	BE	CE	EE	IE	Average
2	52,6	42,25	58,73	63,34	50,29	52,48 [51,94 : 53,02]
4	47,42	36,95	53,31	58,18	43,28	46,88 [46,11 : 47,64]
10	37,24	30,84	42,17	49,28	32,45	37,73 [36,56 : 38,91]
30	16,02	16,39	22,34	29,14	14,88	19,49 [17,88 : 21,21]
60	9,38	9,66	11,03	15,83	4,86	10,09 [8,44 : 12,03]
120	1,56	0,97	1,39	8,7	0,0	2,37 [1,38 : 4,03]
300	0,0	0,0	0,0	7,41	0,0	1,33 [0,44 : 3,95]

Table B.24: *X2_CSV*M configuration *RER* [%].

Test segment length [s]	AE	BE	CE	EE	IE	Average
2	54,35	40,43	57,87	63,25	49,53	51,95 [51,41 : 52,49]
4	47,48	36,47	52,03	57,12	42,69	46,21 [45,44 : 46,97]
10	35,16	28,59	43,09	46,64	31,3	36,24 [35,09 : 37,42]
30	15,63	15,42	20,62	26,62	12,8	17,99 [16,43 : 19,66]
60	8,59	9,18	12,41	20,14	2,78	10,48 [8,8 : 12,45]
120	0,0	1,94	0,0	5,8	0,0	1,58 [0,82 : 3,02]
300	0,0	0,0	0,0	7,41	0,0	1,33 [0,44 : 3,95]

Table B.25: *C2* configuration *RER* [%].

Test segment length [s]	AE	BE	CE	EE	IE	Average
2	52,47	41,2	57,89	62,89	49,32	51,75 [51,2 : 52,29]
4	47,68	36,41	52,35	57,65	42,23	46,29 [45,53 : 47,06]
10	35,81	30,28	40,57	47,12	30,61	36,29 [35,13 : 37,46]
30	17,19	15,66	23,02	27,34	13,84	19,1 [17,5 : 20,81]
60	8,59	9,66	8,97	15,11	4,17	9,31 [7,72 : 11,18]
120	1,56	0,97	1,39	7,25	0,0	2,11 [1,19 : 3,7]
300	0,0	0,0	0,0	7,41	0,0	1,33 [0,44 : 3,95]

Table B.26: *C2_CSV*M configuration *RER* [%].

Test segment length [s]	AE	BE	CE	EE	IE	Average
2	54,89	39,16	56,36	62,26	48,88	51,11 [50,57 : 51,65]
4	47,84	34,48	51,53	55,54	41,95	45,21 [44,44 : 45,97]
10	36,33	26,67	41,94	45,2	29,23	35,05 [33,9 : 36,21]
30	17,58	14,7	20,96	26,26	11,07	17,79 [16,24 : 19,45]
60	9,38	7,25	8,97	16,55	2,78	8,78 [7,24 : 10,62]
120	0,0	0,97	0,0	5,8	0,0	1,32 [0,64 : 2,68]
300	0,0	0,0	0,0	7,41	0,0	1,33 [0,44 : 3,95]

Table B.27: *D2 configuration RER [%].*

Test segment length [s]	AE	BE	CE	EE	IE	Average
2	52,86	42,22	58,51	63,94	50,47	52,61 [52,07 : 53,15]
4	47,11	37,02	53,17	58,8	43,97	47,06 [46,29 : 47,83]
10	36,46	31,24	42,17	48,92	32,57	37,66 [36,49 : 38,84]
30	16,8	16,39	22,34	28,06	13,84	19,23 [17,63 : 20,94]
60	7,81	10,14	9,66	14,39	4,86	9,44 [7,84 : 11,32]
120	1,56	0,97	1,39	8,7	0,0	2,37 [1,38 : 4,03]
300	0,0	0,0	0,0	7,41	0,0	1,33 [0,44 : 3,95]

Table B.28: *D2_CSVM configuration RER [%].*

Test segment length [s]	AE	BE	CE	EE	IE	Average
2	54,58	40,34	57,0	64,04	49,11	51,86 [51,32 : 52,41]
4	47,32	36,15	51,94	57,94	42,64	46,22 [45,45 : 46,98]
10	36,2	28,84	42,97	46,76	30,96	36,42 [35,26 : 37,6]
30	15,63	15,9	22,68	26,26	12,11	18,31 [16,74 : 20,0]
60	8,59	8,7	11,03	20,14	2,78	10,09 [8,44 : 12,03]
120	0,0	1,94	0,0	5,8	0,0	1,58 [0,82 : 3,02]
300	0,0	0,0	0,0	7,41	0,0	1,33 [0,44 : 3,95]

Table B.29: *X3 configuration RER [%].*

Test segment length [s]	AE	BE	CE	EE	IE	Average
2	54,22	40,31	58,44	62,46	49,8	51,91 [51,37 : 52,45]
4	47,01	36,05	52,86	55,4	43,33	45,98 [45,22 : 46,75]
10	35,68	28,59	43,09	45,44	30,38	35,94 [34,78 : 37,11]
30	16,8	16,39	20,96	26,26	12,46	18,38 [16,81 : 20,06]
60	7,03	8,21	10,34	17,27	2,78	9,04 [7,48 : 10,9]
120	0,0	1,94	0,0	7,25	0,0	1,84 [1,0 : 3,36]
300	0,0	0,0	0,0	7,41	0,0	1,33 [0,44 : 3,95]

Figure B.23: *X2* configuration RER [%].

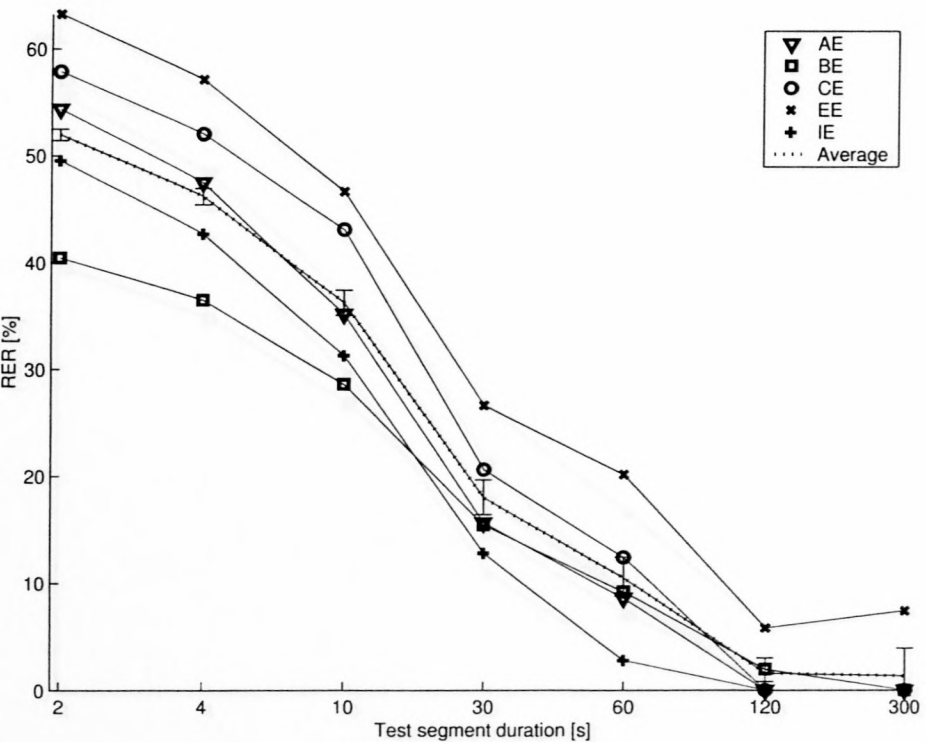
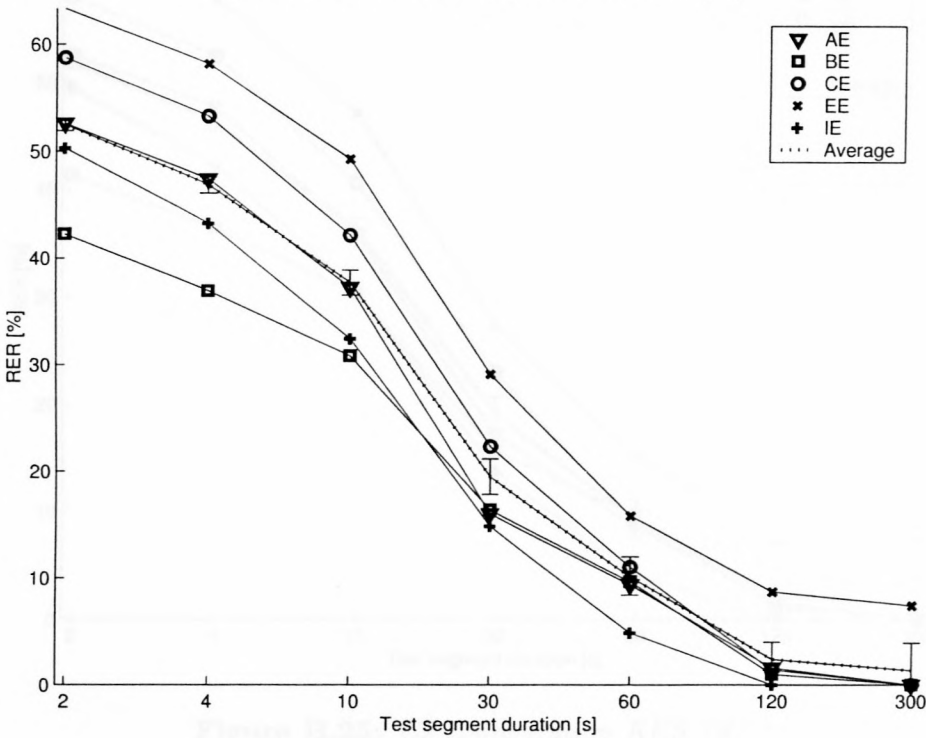


Figure B.24: *X2_CSVM* configuration RER [%].

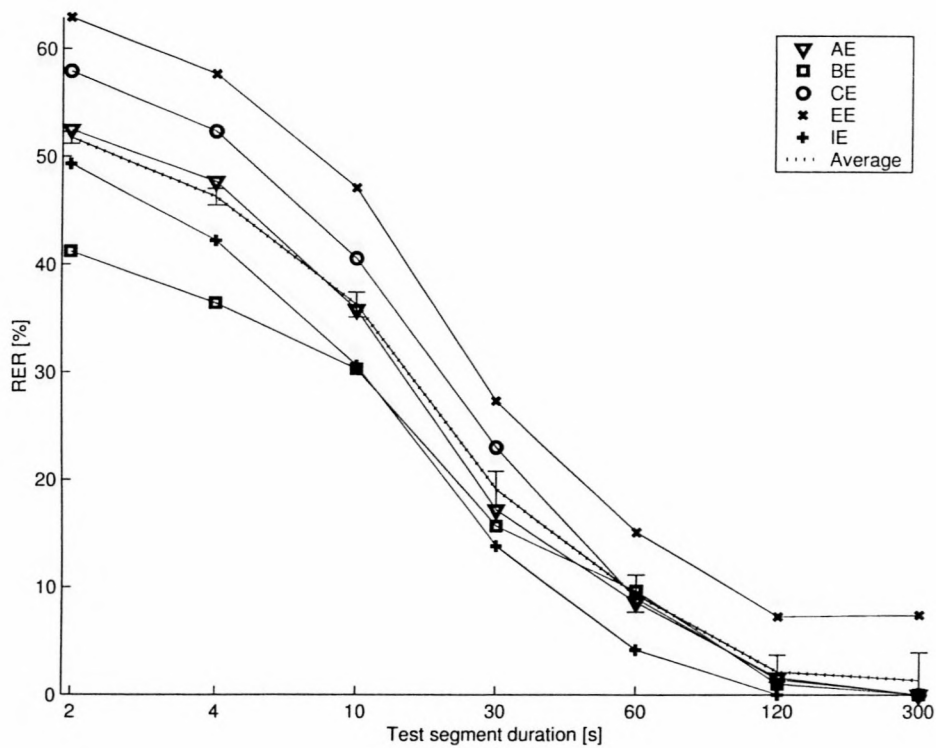


Figure B.25: C2 configuration RER [%].

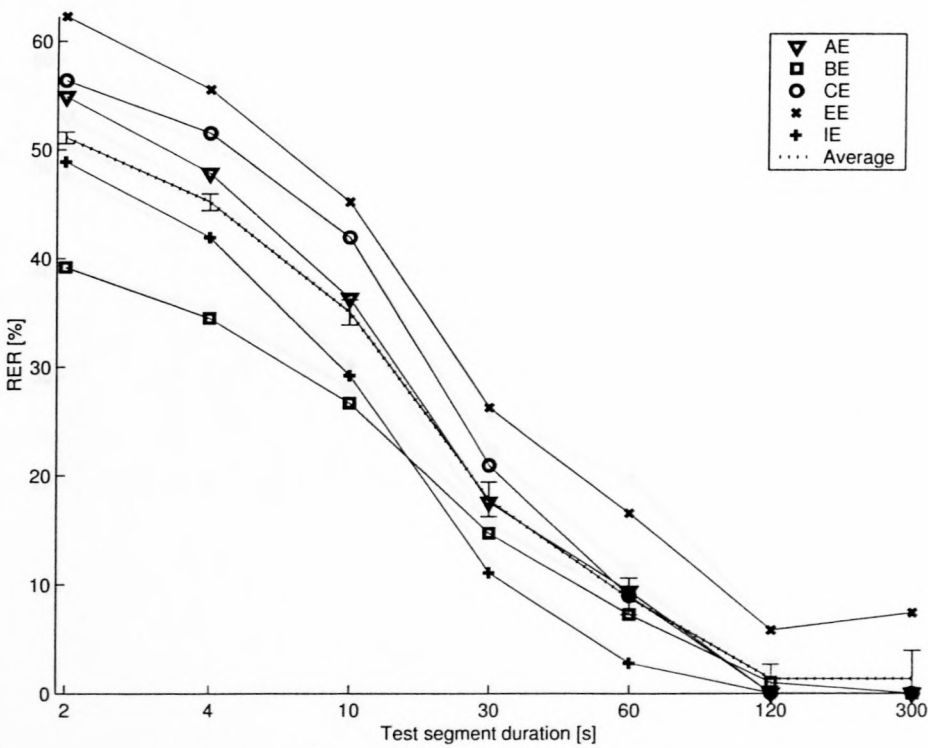


Figure B.26: C2_CSV configuration RER [%].

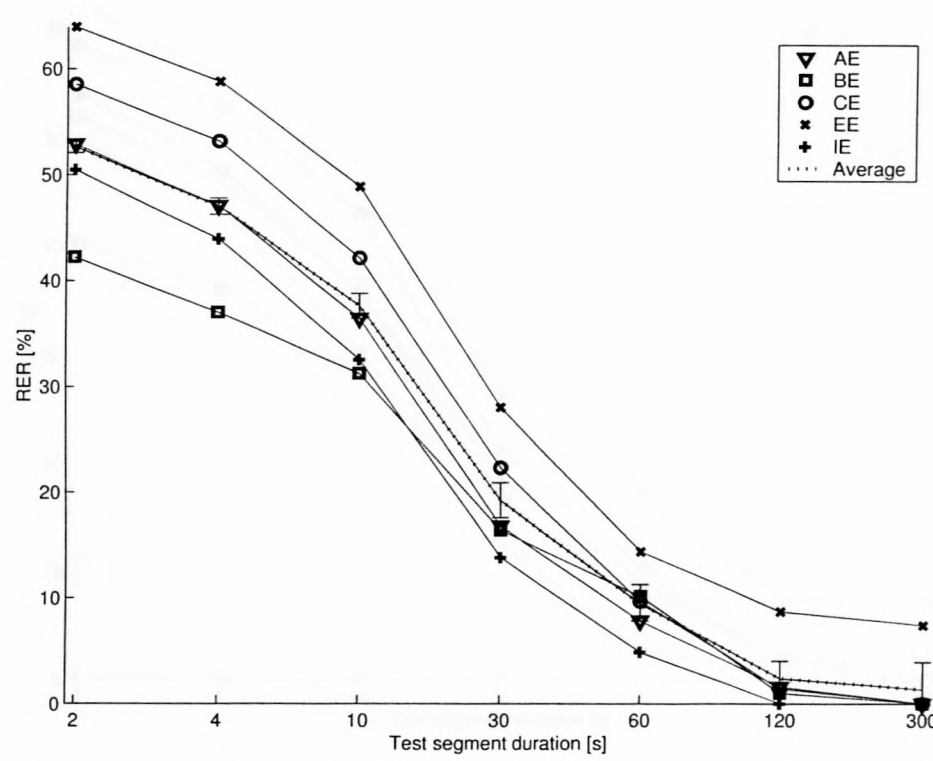


Figure B.27: *D2* configuration RER [%].

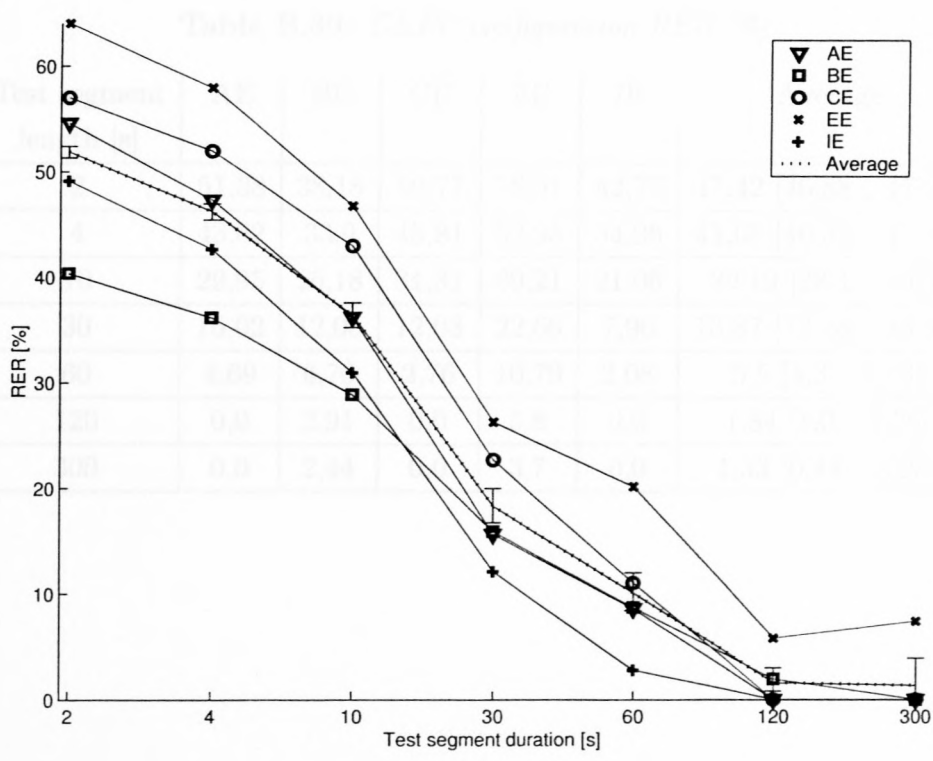


Figure B.28: *D2_CSV* configuration RER [%].

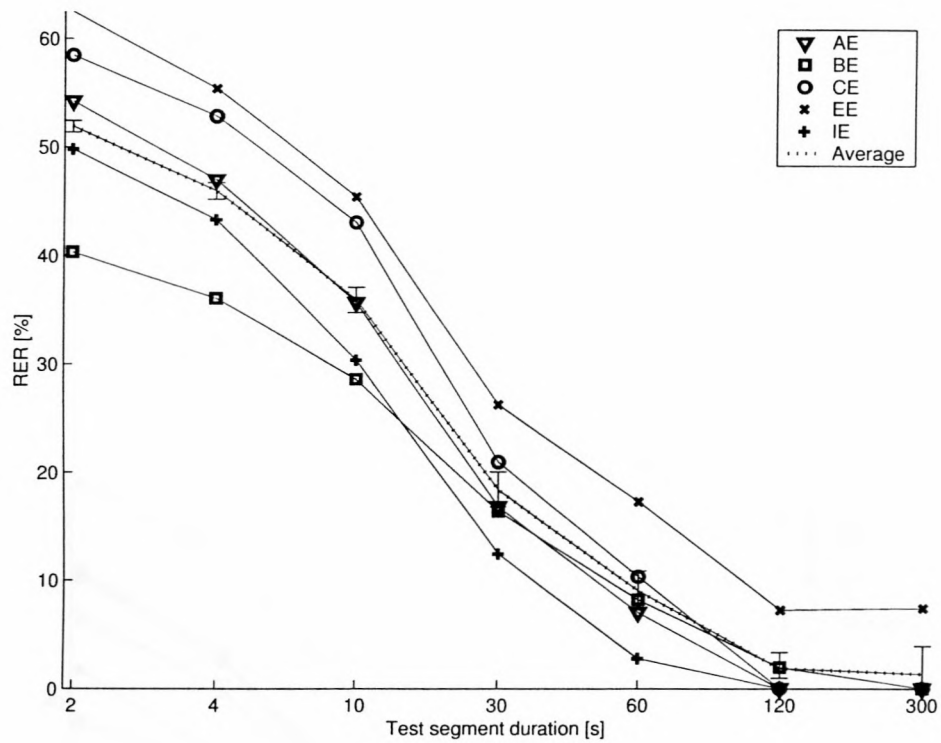


Figure B.29: *X3 configuration RER [%].*

Table B.30: *C2_FC configuration RER [%].*

Test segment length [s]	AE	BE	CE	EE	IE	Average
2	51,38	38,18	50,77	58,91	42,77	47,42 [46,88 : 47,96]
4	43,62	33,9	43,81	52,95	34,96	41,08 [40,32 : 41,83]
10	29,95	26,18	31,31	39,21	21,06	29,19 [28,1 : 30,3]
30	16,02	12,05	12,03	22,66	7,96	13,87 [12,48 : 15,38]
60	4,69	6,76	2,76	10,79	2,08	5,5 [4,3 : 7,03]
120	0,0	2,91	0,0	5,8	0,0	1,84 [1,0 : 3,36]
300	0,0	2,44	0,0	3,7	0,0	1,33 [0,44 : 3,95]

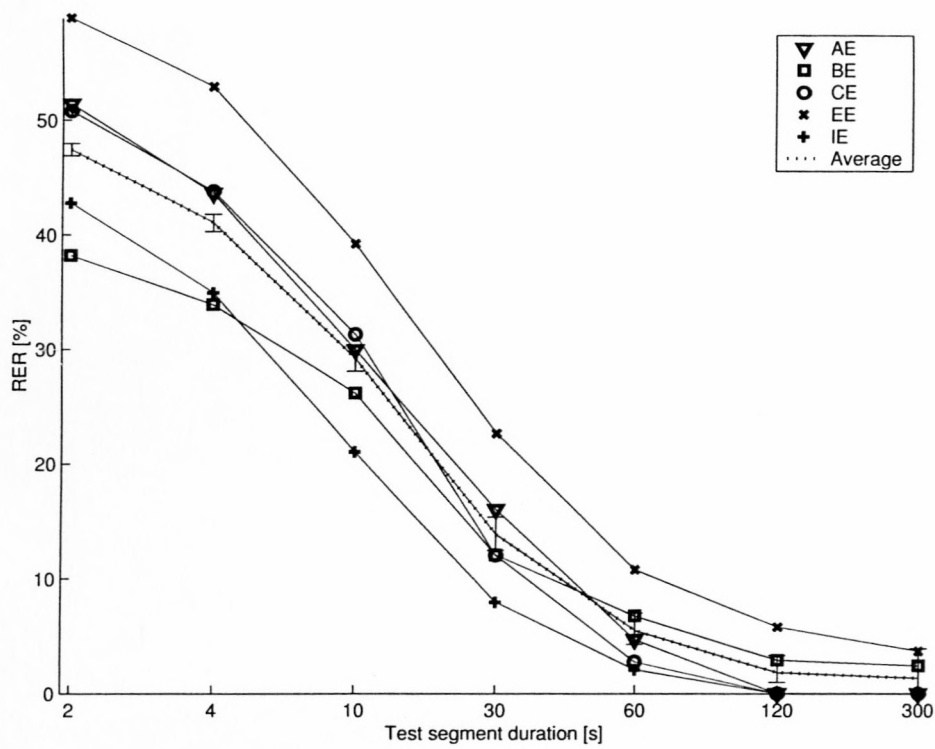


Figure B.30: *C2_FC* configuration RER [%].



dutoit_automatic_2004

